

CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL
UNIDAD ZACATENCO
DEPARTAMENTO DE COMPUTACIÓN

**“Análisis de los dispositivos Space Navigator, Myo y
Leap Motion y su evaluación en aplicaciones de
software”**

T E S I S

Que presenta

AXEL SALAZAR ORDOÑEZ

Para obtener el grado de

Maestro en Ciencias en Computación

Director de la Tesis: **Dr. Luis Gerardo de la Fraga**

Ciudad de México

Septiembre, 2016

AGRADECIMIENTOS

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo económico durante estos dos años.

RESUMEN

Hoy en día hay un gran interés por realizar investigación en la interacción con las computadoras a través de dispositivos basados en el control por gestos. Los dispositivos Leap Motion y Myo, están siendo sujetos a investigación como medios de interacción con aplicaciones de software en diferentes áreas. No cabe duda que con este tipo de dispositivos aumentan las posibilidades de desarrollar interfaces de usuario más naturales, en comparación con dispositivos como el ratón y el teclado convencionales, pero al ser dispositivos recientes, primero se necesita realizar un análisis de sus características y capacidades. Es por esto que en esta tesis se presenta el estudio, análisis y evaluación de las capacidades como la precisión y repetibilidad de los dispositivos Leap Motion, Myo y Space Navigator. Además se presenta el análisis y evaluación del desempeño de los dispositivos en aplicaciones en 2D y 3D, utilizando la Ley de Fitts y el estándar ISO 9241-9 como marcos de trabajo. Se dan sugerencias sobre el uso de estos dispositivos en aplicaciones donde resulta adecuada su utilización, tanto en aplicaciones 2D, como en 3D, con base en los resultados obtenidos de los experimentos realizados. El dispositivo Leap Motion se puede usar en aplicaciones 3D, pero no para seleccionar objetos pequeños. Se debe evitar usarlo por periodos de tiempo prolongados ya que causa mucha fatiga en el brazo. El dispositivo Myo se puede usar en aplicaciones 2D, tal como se usó en esta tesis (integrando la velocidad angular de su giróscopo). El dispositivo Space Navigator es más recomendable usarlo en aplicaciones 3D.

ABSTRACT

Nowadays there is a strong interest in conducting research on the interaction with computers through devices based on the control by gestures. The devices Leap Motion and Myo are being subjects to research as a means of interacting with software applications in different areas. There is no doubt that these devices increase the chances of developing more natural user interfaces, compared with devices such as the standard mouse and keyboard, but being too recent devices, we first need to conduct an analysis of its features and capabilities. That is why in this thesis, the study, analysis, and evaluation of the capabilities as accuracy and repeatability of the Leap Motion, Myo and Space Navigator devices are presented. In addition, the analysis and evaluation of the performance of these devices in both, 2D and 3D applications are presented using as frameworks the Fitts's Law and the ISO 9241-9 standard. Based on the results of the conducted experiments, we suggest for which 2D or 3D applications are more suitable the use of these devices. The Leap Motion device can be used in 3D applications, but not to select small objects. Also, it is avoided to use this device by long periods of time because produce arm fatigue. Myo device can be used in 2D applications, in the way as it is used in this thesis (by integration of the angular velocity of its gyroscope). Space Navigator is recommended to use it in 3D applications.

ÍNDICE GENERAL

Índice general	VII
Lista de Figuras	X
Lista de Tablas	XII
1 Introducción	1
1.1. Planteamiento del problema	2
1.2. Objetivos de la tesis	3
1.2.1. General	3
1.2.2. Particulares	3
1.3. Estado del arte	3
1.4. Organización de la tesis	5
2 Análisis del dispositivo Leap Motion	7
2.1. Descripción del dispositivo	7
2.2. Descripción del experimento	8
2.2.1. Configuración de las mediciones	8
2.2.2. Metrología	11
2.2.3. Experimentos estático y dinámico	12
2.3. Resultados	12
2.3.1. Mediciones del modelo de la mano de plástico	13
2.3.1.1. Escenario estático	13
2.3.1.2. Escenario dinámico	15
2.3.2. Mediciones usando un lápiz	17
2.3.2.1. Escenario estático	17
2.3.2.2. Escenario dinámico	17
3 Análisis del dispositivo Myo	19
3.1. Descripción del dispositivo	19

3.2.	Gestos que reconoce el dispositivo	20
3.3.	Uso del giróscopo para controlar un desplazamiento en 3D	21
3.3.1.	Filtrar los valores que entrega el giróscopo	21
3.3.2.	Convertir los datos del giróscopo al marco de referencia del mundo	22
3.3.3.	Extraer los parámetros x , y y z del vector con los datos del giróscopo	22
3.3.4.	Determinar la ganancia	23
3.4.	Experimentos	23
4	Análisis del dispositivo Space Navigator	25
4.1.	Descripción del dispositivo	25
4.2.	Implementación del manejador del dispositivo	26
4.2.1.	Tipos de manejadores en los sistemas Unix	26
4.2.2.	Comunicación dispositivo-computadora	27
4.2.2.1.	Abrir conexión con el dispositivo	27
4.2.3.	Lectura de los valores que entregan los sensores	28
4.2.4.	Interpretación de los valores entregados por los sensores	28
4.2.4.1.	Valores de la velocidad de traslación y rotación	29
4.3.	Descripción del experimento	30
4.3.1.	Configuración de las mediciones	30
4.3.2.	Experimentos	31
4.3.3.	Análisis estadístico	31
4.4.	Resultados	31
5	Uso de los dispositivos en una aplicación para evaluar su desempeño	35
5.1.	Metodología	36
5.1.1.	Participantes	37
5.1.2.	Equipo utilizado	37
5.2.	Descripción de la aplicación en 2D	37
5.2.1.	Forma de usar la aplicación 2D	38
5.2.2.	Resultados	39
5.2.2.1.	Tiempo del movimiento promedio	40
5.2.2.2.	Rendimiento	42
5.2.2.3.	Tasa de error	42
5.2.2.4.	Usabilidad	44
5.3.	Descripción de la aplicación en 3D	45
5.3.1.	Forma de usar la aplicación 3D	46
5.3.2.	Resultados	47
5.3.2.1.	Tiempo del movimiento	47
5.3.2.2.	Usabilidad	49
6	Conclusiones	51
6.1.	Leap Motion	51
6.2.	Myo	52
6.3.	Space Navigator	53
6.4.	Conclusiones finales	54

6.5. Trabajo futuro	55
A Código para usar el giróscopo del Myo como ratón	57
B Código del manejador del dispositivo Space Navigator	61
Bibliografía	67

LISTA DE FIGURAS

1.1. Configuración utilizada en [1]	4
1.2. Configuraciones utilizadas en [2]	4
2.1. Vista de los componentes internos (a) y del campo de visión de 150° (b) del dispositivo.	8
2.2. Configuración utilizada para las mediciones de los dos objetos de referencia, sobre el eje x (a) modelo de la mano y (b) lápiz, sobre el eje y (c) modelo de la mano y (d) lápiz, sobre el eje z (e) modelo de la mano y (f) lápiz	9
2.3. (a) Circuito realizado sobre dos placas de prueba o <i>protoboards</i> , (b) Fuente de poder utilizada para alimentar el circuito	10
2.4. Interfaz gráfica de la aplicación desarrollada en Qt para la automatización de las mediciones.	10
2.5. Gráfica de la precisión de la mano de plástico sobre los ejes x , y y z , en el escenario estático.	14
2.6. Gráfica de la repetibilidad de la mano de plástico sobre los ejes x , y y z , en el escenario estático.	14
2.7. Gráfica de la precisión de la mano de plástico sobre los ejes x , y y z , en el escenario dinámico.	16
2.8. Gráfica de la repetibilidad de la mano de plástico sobre los ejes x , y y z , en el escenario dinámico.	16
2.9. Precisión (a) y repetibilidad (b) del lápiz sobre los ejes x , y y z , en el escenario estático.	17
2.10. Precisión (a) y repetibilidad (b) del lápiz sobre los ejes x , y y z , en el escenario dinámico.	18
3.1. Dispositivo Myo. Es una banda que se coloca alrededor del antebrazo.	20
3.2. Gestos predeterminados que reconoce el dispositivo. (a) <i>tic doble</i> , (b) <i>puño</i> , (c) <i>dedos abiertos</i> , (d) <i>onda izquierda</i> y (e) <i>onda derecha</i>	20
3.3. Sistema de coordenadas del dispositivo Myo.	22
4.1. Dispositivo Space Navigator.	26
4.2. Configuración del experimento para el análisis del dispositivo Space Navigator.	30
4.3. Gráfica de la media +/- y de la desviación estándar de los valores obtenidos del experimento.	33
5.1. Las siete condiciones de la aplicación en 2D.	39

5.2. Gráfica de las medias de los tiempos de movimiento (t_m) del dispositivo Leap Motion.	40
5.3. Gráfica de las medias de los tiempos de movimiento (t_m) del dispositivo Myo.	41
5.4. Gráfica de las medias de los tiempos de movimiento (t_m) del dispositivo Space Navigator.	41
5.5. Gráfica de la media del rendimiento (r) de los dispositivos por cada índice de dificultad (I_d).	42
5.6. Gráfica de la tasa de error de los dispositivos por cada índice de dificultad (I_d).	43
5.7. (a) Posición inicial, las dos teteras están superpuestas, (b) La tetera de color verde (<i>objetivo</i>), es desplazada una distancia D en la dirección 2 (hacia la esquina inferior izquierda). (c) Posición inicial, misma situación que en (a), (d) La tetera de color verde (<i>objetivo</i>), es desplazada una distancia D en la dirección 1 (hacia la esquina superior derecha)	47
5.8. Gráfica de las medias de los tiempos de movimiento (t_m) de los dispositivos por cada índice de dificultad (I_d) en la dirección 1 en la aplicación en 3D.	48
5.9. Gráfica de las medias de los tiempos de movimiento (t_m) de los dispositivos por cada índice de dificultad (I_d) en la dirección 2 en la aplicación en 3D.	49

LISTA DE TABLAS

2.1. Precisión y repetibilidad en los tres ejes cartesianos	13
2.2. Precisión y repetibilidad en los 3 ejes cartesianos	15
2.3. Precisión y repetibilidad en los 3 ejes cartesianos usando un lápiz (escenario estático).	17
2.4. Precisión y repetibilidad en los 3 ejes cartesianos	18
4.1. Valores de los bytes de la lista de caracteres.	29
4.2. Valores y significado de los bytes de desbordamiento.	29
4.3. Valores obtenidos de las mediciones sobre el eje x positivo.	32
4.4. Valores obtenidos de las mediciones sobre el eje x negativo.	32
4.5. Valores obtenidos de las mediciones sobre el eje z positivo.	32
4.6. Valores obtenidos de las mediciones sobre el eje z negativo.	33
5.1. Condiciones del experimento para la evaluación de la Ley de Fitts en la aplicación 2D	38
5.2. Media de los tiempos de movimiento (t_m) de los dispositivos por cada índice de dificultad (I_d).	40
5.3. Valores de las constantes a y b para cada dispositivo, así como su error estándar asintótico, calculados con regresión lineal con gnuplot.	41
5.4. Media del rendimiento (r) de los dispositivos por cada índice de dificultad (I_d).	42
5.5. Tasa de error de los dispositivos por cada índice de dificultad (I_d).	43
5.6. Valores de las medias de las respuestas al cuestionario realizado a los seis participantes.	44
5.7. Medias del tiempo de movimiento (t_m), rendimiento (r) y tasa de error de los dispositivos en la aplicación en 2D.	45
5.8. Condiciones del experimento para la evaluación de la Ley de Fitts en la aplicación 3D	45
5.9. Media del tiempo de movimiento (t_m) de los dispositivos por cada índice de dificultad (I_d) en la dirección 1 en la aplicación en 3D.	48
5.10. Media del tiempo de movimiento (t_m) de los dispositivos por cada índice de dificultad (I_d) en la dirección 2.	48
5.11. Valores de las medias de las respuestas al cuestionario realizado a los cinco participantes.	50
5.12. Medias del tiempo de movimiento (t_m) de los dispositivos en la aplicación en 3D.	50
6.1. Tabla comparativa de los tres dispositivos. 1 = Mejor, 2 = regular y 3 = peor	54

1

INTRODUCCIÓN

Durante más de 30 años se ha estudiado la interacción con las computadoras a través de gestos. Dispositivos como el mando de la consola Wii de Nintendo [3] y el Kinect de Microsoft Xbox [4], han sido algunos de los resultados de estas investigaciones. Este tipo de dispositivos están basados en el reconocimiento de gestos, mismos que son captados por medio de sensores y/o cámaras. El reconocimiento de gestos se refiere al reconocimiento de expresiones significativas del movimiento por un ser humano, el cual puede involucrar los dedos, las manos, los brazos, la cara, la cabeza y/o el cuerpo [5]. Como se menciona en [5], los gestos pueden ser estáticos (el usuario adopta una pose o una configuración) o dinámicos (con movimiento). El avance en áreas como el procesamiento de imágenes, visión por computadora y desarrollo de sensores, ha jugado un rol importante en la manera de hacer el reconocimiento de gestos. Dicho reconocimiento va desde sensores de rayos infrarrojos y cámaras, hasta avanzadas técnicas de visión [6] y detección de señales eléctricas (electromiografía).

Por otro lado, las interfaces basadas en el concepto de Interfaz de Usuario Natural (NUI) por sus siglas en inglés [7], están tomando una mayor relevancia debido a la introducción de dispositivos de control por gestos como: Leap Motion, Myo, Kinect, etc. Esto hace que los usuarios puedan interactuar con estas aplicaciones de una manera más intuitiva, haciéndolas más fáciles de aprender, debido a que lo que se busca es que el usuario haga uso de gestos naturales y cotidianos para operar estas aplicaciones.

En [8], se menciona una lista de las oportunidades para emplear dispositivos basados en el control por gestos, la cual se retoma a continuación:

- **Entretenimiento:** Wii [3] y Kinect [4] han mostrado una forma más natural de realizar cierto tipo de actividades, tales como la selección de música y videos, la personalización y control del juego, etc.

- **Inteligencia artificial:** la industria robótica está usando esta tecnología para gestionar y controlar las actividades del robot como parte de la interacción Humano-Robot.
- **Simulación:** aplicaciones de realidad virtual han sido desarrolladas para la rehabilitación de pacientes. Por ejemplo, la compañía Gesture [9] ha desarrollado varias aplicaciones de este tipo. Así mismo en [10] se muestra una aplicación de terapia de mano.
- **Entrenamiento y educación:** se puede entrenar a las personas automáticamente con base en el perfil del usuario. La gran ventaja con respecto al teclado y al ratón, es que se puede obtener la entrada natural de los movimientos del cuerpo.
- **Vida asistida por computadora:** gente con alguna discapacidad o de edad avanzada, se pueden ver beneficiados con esta tecnología, la cual les pueda dar la oportunidad de adquirir cierta independencia al realizar sus actividades. En [11] realizaron una aplicación de este tipo.

Su gran aplicabilidad nos muestra la importancia de realizar más investigación en torno a los sistemas controlados por gestos, así como su integración con Interfaces de Usuario Naturales (NUI).

1.1 Planteamiento del problema

Hoy en día, hay un gran número de trabajos dedicados a la investigación en la integración de dispositivos como Kinect, Myo, Leap Motion, el mando de la consola Wii, entre otros, con las aplicaciones de software actuales. Nymoen *et al.* [12] exploran la posibilidad de la integración del dispositivo Myo de Thalmic Labs, en la interacción con aplicaciones de expresión musical. En [13] se presenta un prototipo de realidad virtual en el cual se hace uso de los dispositivos Myo y Kinect para el entrenamiento de personas a las cuales se les ha amputado algún brazo. En [10], Lipovský y Ferreira, muestran un prototipo de una aplicación para la rehabilitación de la mano, en donde utilizan el dispositivo Myo en combinación con un guante robótico. Marin *et al.* en [14], proponen un sistema que permite el reconocimiento de gestos del lenguaje americano de señas utilizando los dispositivos Kinect y Leap Motion. En [15], exploran la interacción del dispositivo Myo con un SmartWatch en un mismo brazo, en donde la aplicación que se prueba es un reproductor de música. Una aplicación para el entrenamiento de pacientes con dislexia es propuesta en [16], en donde utilizan el dispositivo Leap Motion.

Debido a la rápida propagación de la utilización de estos dispositivos en una amplia gama de aplicaciones de software, en diferentes áreas, analizar las capacidades de este tipo de dispositivos, es un tema de gran relevancia, ya que debido a la naturaleza de cada una de las aplicaciones, los requerimientos en los dispositivos de interacción con éstas, pueden ser mayores que en otras. Por ejemplo, en el caso de una aplicación para ayudar a la rehabilitación de los pacientes, la precisión del dispositivo debe ser mayor comparada

con aplicaciones sencillas y de otra área, como por ejemplo, un juego de video.

Explorar estos dispositivos y determinar sus alcances, sirve para situarlos en un cierto rango de aplicaciones en las que resulta adecuada su utilización, debido a sus capacidades, y más importante aún, determinar en cuáles no deben de utilizarse debido a las limitaciones de los mismos.

En esta tesis se realizó la evaluación de las capacidades de los dispositivos Leap Motion, Myo y Space Navigator, con el fin de analizar qué tipo de aplicaciones se pueden desarrollar soportando estos dispositivos como forma de interacción. Así mismo, se desarrollaron dos aplicaciones basadas en Qt y OpenGL para medir el desempeño, de acuerdo con la Ley de Fitts, de estos dispositivos con aplicaciones 2D y 3D.

La hipótesis de investigación para el proyecto de tesis es que no resulta adecuado utilizar los dispositivos Leap Motion, Myo y Space Navigator, en todo tipo de aplicaciones, debido a las características y limitaciones de cada uno de ellos.

1.2 Objetivos de la tesis

1.2.1. General

Evaluar las capacidades de los dispositivos Leap Motion, Myo y Space Navigator, con el fin de poder determinar en qué tipo de aplicaciones resulta adecuada su utilización. Así como analizar el desempeño de los mismos en la interacción con aplicaciones en 2D y 3D.

1.2.2. Particulares

1. Crear un manejador para el dispositivo Space Navigator con el fin de poder utilizar el dispositivo en las aplicaciones que se desarrollarán.
2. Estudiar y analizar los dispositivos Space Navigator, Myo y Leap Motion, para poder diseñar de manera correcta los experimentos para su evaluación.
3. Crear dos aplicaciones basadas en Qt y OpenGL, una en 2D y la otra en 3D, para mostrar el desempeño de los dispositivos en este tipo de aplicaciones.
4. Identificar objetivamente en qué tipo de aplicaciones pueden utilizarse estos dispositivos para poder determinar el alcance de los mismos.

1.3 Estado del arte

Weichert *et al.* [1] hacen el análisis de la precisión y repetibilidad del dispositivo Leap Motion, utilizando un robot industrial con una pluma de referencia, lo cual les brinda una precisión de la posición de 0.2 mm. En la figura 1.1, se muestra la configuración que utilizaron. Los resultados que obtuvieron, tanto en configuraciones estáticas, como en

configuraciones dinámicas, están dentro de lo que especifica el fabricante, el cual declara una precisión milimétrica. Es importante mencionar que utilizaron una versión preliminar del dispositivo Leap Motion y que como ellos mencionan, la precisión y repetibilidad pueden ser diferentes en la versión final del dispositivo.

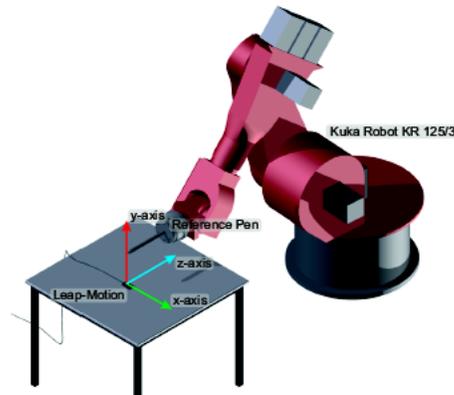


Figura 1.1: Configuración utilizada en [1]

Por otro lado, en [2], para hacer la evaluación del desempeño del mismo dispositivo (Leap Motion), utilizan el sistema de captura de movimiento Qualisys™ como sistema de referencia, así como un modelo de brazo de plástico para hacer la medición en condiciones estáticas y dos palitos de madera con forma de V con marcadores fijos para las condiciones dinámicas. Para las mediciones dinámicas, los objetos fueron movidos aproximadamente con una velocidad constante de 100 mm/s.



Figura 1.2: Configuraciones utilizadas en [2]

En este análisis, concluyen que el dispositivo presenta una frecuencia de muestreo inconsistente, razón por la cual no puede ser utilizado como un sistema profesional de seguimiento o rastreo.

En [17], realizaron el estudio del dispositivo Leap Motion desde el punto de vista del desempeño del usuario en tareas cotidianas como la selección en una interfaz gráfica, comparado con un ratón estándar. Este trabajo está sustentado bajo el marco de la Ley de Fitts, el cual es un modelo descriptivo del movimiento humano utilizado principalmente en las áreas de HCI y ergonomía. Este trabajo presenta diversas limitaciones, por lo que los resultados obtenidos no pueden ser tomados como definitivos.

En lo que respecta al dispositivo Myo de Thalmic Labs, no se han encontrado estudios semejantes sobre la evaluación de sus capacidades. Nymoen *et al.* [12] evalúan y exploran el dispositivo para la interacción musical. Aunque sus resultados son prometedores, se necesita realizar un estudio más exhaustivo para poder determinar de una manera más precisa sus capacidades.

El dispositivo Space Navigator ha sido menos explorado. Sólo en [18], Petrasova *et al.*, realizan una comparación del dispositivo con otros, como el mando del Xbox 360, el mando del Wii, y el teclado y ratón tradicionales, para la interacción con una aplicación de realidad virtual. La forma en que miden el desempeño es a través de un test, en el que se les pregunta a los usuarios qué tan cómodos se sintieron con cada uno de los dispositivos.

Como se puede ver, hace falta que se realicen más y mejores pruebas de los dispositivos, para poder determinar con certeza las capacidades de cada uno. Esto, con el fin de poder ver el alcance de los mismos, en las diferentes y muy variadas áreas posibles de aplicación.

1.4 Organización de la tesis

La presente tesis consta de 6 capítulos, aparte de éste de Introducción. Los seis capítulos restantes están organizados de la siguiente manera: los capítulos 2 a 4 contienen los análisis de los dispositivos Leap Motion, Myo y Space Navigator, respectivamente. En el capítulo 5 se hace el análisis de los tres dispositivos en aplicaciones en 2D y 3D, utilizando la Ley de Fitts. Finalmente, en el capítulo 6 se concluye este trabajo y se plantea algún trabajo futuro que aún podría realizarse.

2

ANÁLISIS DEL DISPOSITIVO LEAP MOTION

En el presente capítulo se detallará el proceso que se llevó a cabo para realizar el análisis de la precisión y repetibilidad del dispositivo Leap Motion. Para esto, se tomaron mediciones del dispositivo en condiciones estáticas y dinámicas del modelo de una mano de plástico, así como también de un lápiz. En la sección 2.1, se realiza la descripción del dispositivo Leap Motion, de como funciona, tanto de los sensores que lo componen. En la sección 2.2, se describe el experimento realizado, se describen la infraestructura, configuraciones y el sistema metrológico utilizados. La sección 2.3 muestra los resultados obtenidos del experimento, tanto de las mediciones del modelo de la mano de plástico, como de las mediciones usando un lápiz, esto en ambos escenarios, estático y dinámico.

2.1 Descripción del dispositivo

El dispositivo Leap Motion reconoce y realiza un seguimiento de las manos, dedos y herramientas parecidas a los dedos como una pluma o un lápiz. Brinda información de la posición de estos objetos en su sistema de coordenadas, además de la información acerca de los gestos realizados. El controlador Leap Motion utiliza sensores ópticos y de luz infrarroja. Tiene un campo de visión de 150° , el cual tiene la forma de una pirámide invertida como se muestra en la figura 2.1(b). El software del dispositivo se ejecuta como un servicio o demonio (en mac y linux). El SDK de Leap Motion provee dos interfaces para obtener los datos del dispositivo: una interfaz nativa y una interfaz WebSocket. El dispositivo captura dos imágenes en escala de grises en la frecuencia cercana al infrarrojo, las procesa para obtener las posiciones de la mano y de los dedos. Las dos imágenes son en estéreo y se procesan con algoritmos de Visión por computadora. Entonces el dispositivo procesa estas imágenes en una memoria local y los algoritmos que usa son internos al dispositivo.



Figura 2.1: Vista de los componentes internos (a) y del campo de visión de 150° (b) del dispositivo.

Finalmente los datos de posición se envían a la computadora a través del puerto USB y se obtienen en la aplicación en las APIs que da el fabricante. Estas APIs están disponibles para los sistemas operativos Windows, Mac OS X y GNU/Linux.

2.2 Descripción del experimento

Para medir el desempeño del dispositivo Leap Motion, se utilizó como sistema de referencia un módulo lineal de alta precisión *Bosch Rexroth AG D-97419*, que se muestra en todas las imágenes de la figura 2.2, el cual tiene una precisión de 50 μm (micrómetros) por cada paso del motor a pasos que lo controla. Se tomaron las mediciones de las posiciones de un modelo de una mano de plástico, así como de un lápiz. Esto tanto en condiciones estáticas como dinámicas. La configuración del experimento consiste de un dispositivo Leap Motion y un módulo lineal de alta precisión, así como de un modelo de una mano de plástico y un lápiz como objetos de referencia. Debido a que el módulo lineal tiene un sólo grado de libertad, las mediciones se tuvieron que realizar individualmente por cada eje del espacio cartesiano (x , y y z).

2.2.1. Configuración de las mediciones

Para la configuración de las mediciones estáticas y dinámicas, el modelo de la mano de plástico y el lápiz, se acoplaron a una base de madera la cual se fijó al módulo lineal de alta precisión *Bosch Rexroth AG D-97419* como se muestra en la figura 2.2.

El dispositivo Leap Motion fue colocado en una mesa plana a una altura de 25 cm del módulo lineal. El dispositivo fue perfectamente fijado a la mesa para evitar que se desplazará en alguna dirección debido a algún descuido. Se desarrolló una aplicación de software basada en Qt y C++ para la realización de las mediciones, en la figura 2.4 se muestra la interfaz de usuario. Mediante esta aplicación se controla el módulo lineal y el dispositivo Leap Motion, logrando una perfecta sincronización en las mediciones. Además, se guardaron los datos de las mediciones en archivos de texto plano para posteriormente realizar los respectivos cálculos de las dos mediciones de repetibilidad y precisión. Para el

aseguramiento del buen desempeño del dispositivo Leap Motion, se utilizó la herramienta *Visualizer* para verificar que se realizó la detección de los objetos de manera correcta. La aplicación *Visualizer* es una de las herramientas que están dentro del menú del panel de control de Leap Motion, misma que nos permite ver los datos de seguimiento del movimiento generados por el controlador Leap Motion, permitiéndonos verificar si el dispositivo está detectando correctamente los objetos. Se utilizaron dos equipos de cómputo diferentes, una computadora portátil Intel® Core™ i7-3610QM CPU @ 2.30GHz con 8 GB de RAM y una Macbook pro Intel® Core™ i7 CPU @ 2.9GHz con 8 GB de RAM.

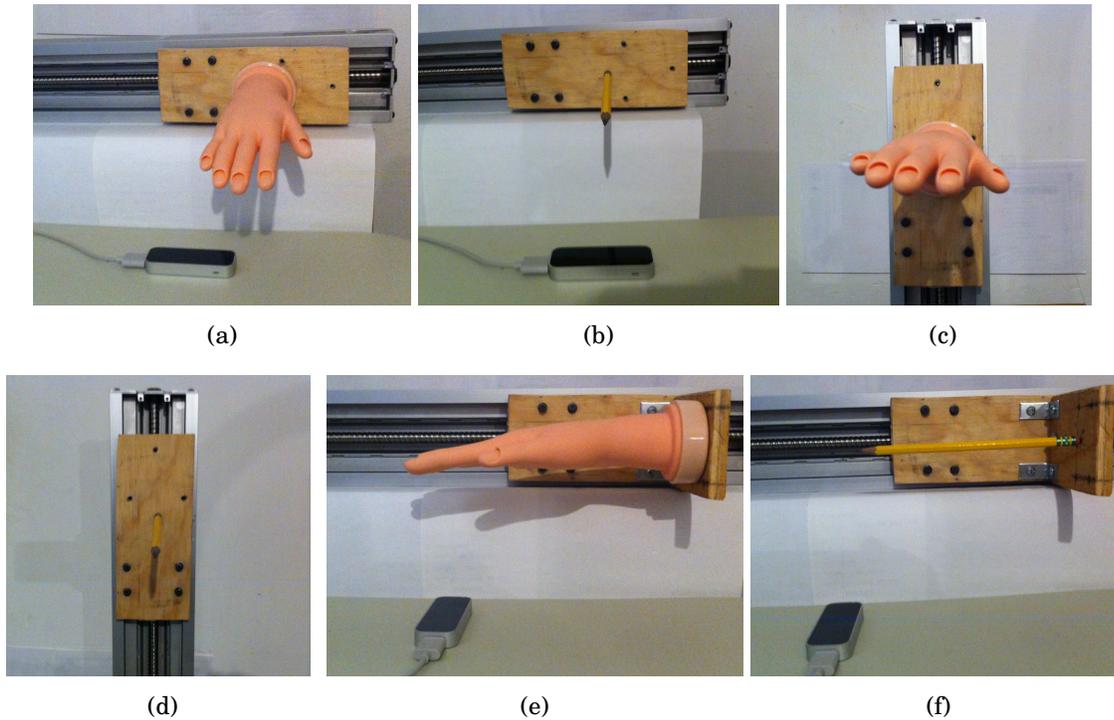


Figura 2.2: Configuración utilizada para las mediciones de los dos objetos de referencia, sobre el eje x (a) modelo de la mano y (b) lápiz, sobre el eje y (c) modelo de la mano y (d) lápiz, sobre el eje z (e) modelo de la mano y (f) lápiz

Para realizar las mediciones de los objetos sobre el eje z se agregó a la base de madera una madera en escuadra ya que el dispositivo Leap Motion presentaba problemas al hacer la detección de los objetos. Para las mediciones sobre el eje y se tuvo que poner en forma vertical el módulo lineal como se muestra en la figura 2.2(c) y 2.2(d).

Para el control del módulo lineal de alta precisión *Bosch Rexroth AG D-97419*, se tuvo que construir un circuito electrónico basado en el diseño de [19], con unas modificaciones debido a las condiciones de software y hardware disponibles para su realización. En la figura 2.3 se muestran los componentes principales de este circuito. La fuente de poder que se usó para alimentar el módulo lineal fue la línea de 12 volts de una fuente conmutada para una PC, como se muestra en la figura 2.3.

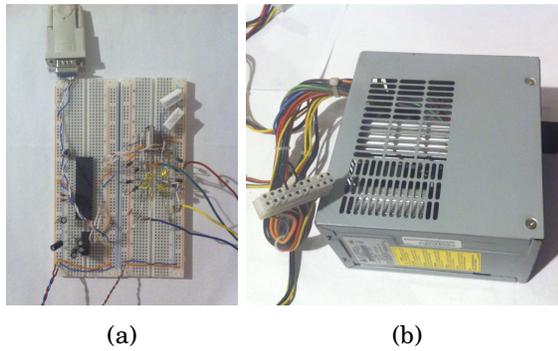


Figura 2.3: (a) Circuito realizado sobre dos placas de prueba o *protoboards*, (b) Fuente de poder utilizada para alimentar el circuito

Se desarrolló una aplicación en Qt para la automatización de las mediciones, en la cual, se puede configurar el eje sobre el que se realizarán las mediciones, el tipo del objeto (mano o lápiz), el escenario (estático o dinámico), el rango de las repeticiones, así como cargar el archivo con las posiciones de referencia.

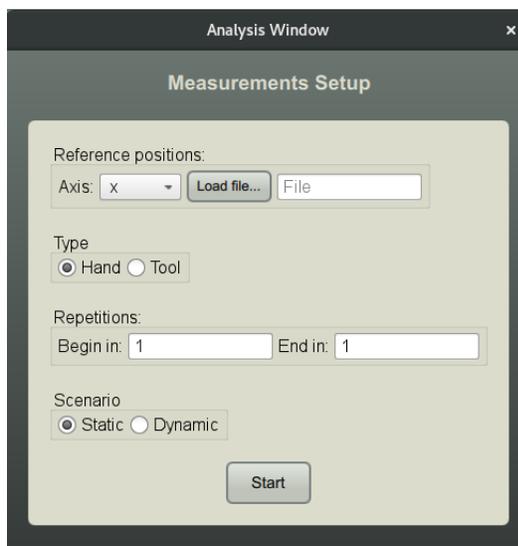


Figura 2.4: Interfaz gráfica de la aplicación desarrollada en Qt para la automatización de las mediciones.

2.2.2. Metrología

El sistema metrológico [20] se definió en base a la configuración del experimento para realizar las mediciones de la repetibilidad y precisión del dispositivo Leap Motion. En [1] se define la precisión como la habilidad de un sensor 3D para determinar una posición deseada en el espacio 3D, y la repetibilidad como la habilidad de un sensor para localizar la misma posición en cada repetición de la medición. Al igual que en [1], se utilizó el estándar ISO 9283 [21], con una adaptación en cuanto al número de ciclos, en lugar de utilizar los 30 ciclos que se define en el documento, se utilizaron 200 por cada posición, esto debido a que en [22] Jeswiet y Helferty demostraron que los 30 ciclos sugeridos por el ISO no fueron suficientes para determinar la repetibilidad a largo plazo y que al menos 200 ciclos son necesarios. La precisión p , para una deseada posición 3D x_c es calculada por

$$p = \sqrt{(\bar{x} - x_c)^2 + (\bar{y} - y_c)^2 + (\bar{z} - z_c)^2} \quad (2.1)$$

en donde,

$$\bar{x} = \frac{1}{n} \sum_{j=1}^n x_j, \quad \bar{y} = \frac{1}{n} \sum_{j=1}^n y_j, \quad \bar{z} = \frac{1}{n} \sum_{j=1}^n z_j \quad (2.2)$$

\bar{x} , \bar{y} y \bar{z} son las medias aritméticas de las posiciones obtenidas después de repetir la misma pose n veces. Y x_c , y_c y z_c son las posiciones de referencia. La repetibilidad, r , se calcula como

$$r = \bar{v} - 3s_v \quad (2.3)$$

en donde,

$$\bar{v} = \frac{1}{n} \sum_{j=1}^n v_j, \quad v_j = \sqrt{(x_j - \bar{x})^2 + (y_j - \bar{y})^2 + (z_j - \bar{z})^2} \quad (2.4)$$

representa la varianza de las posiciones medidas y

$$s_v = \sqrt{\frac{\sum_{j=1}^n (v_j - \bar{v})^2}{n - 1}} \quad (2.5)$$

la desviación estándar correspondiente.

En la siguiente sección se presenta a detalle la descripción de los experimentos realizados con el fin de analizar la precisión y repetibilidad del sensor en escenarios estáticos y dinámicos.

2.2.3. Experimentos estático y dinámico

Con el fin de evaluar la precisión y repetibilidad del dispositivo Leap Motion, se tomaron dos objetos como referencias, un modelo de una mano de plástico y un lápiz. En el caso del modelo de la mano de plástico, se tomaron las posiciones de los cinco dedos y de la palma de la mano. Se definieron dos escenarios para realizar las mediciones de ambos objetos, uno estático y otro dinámico. En ambos escenarios, se tomaron un conjunto de 11 posiciones de referencia, las cuales estaban a pocos milímetros de distancia. Como el sistema de referencia utilizado (módulo lineal Rexroth) sólo nos ofrece un grado de libertad, se comenzó y se terminó el ciclo en la posición inicial. Asimismo, debido a esta limitante, se tuvieron que analizar por separado los 3 ejes coordenados.

En el escenario estático, se definió un tiempo de espera entre cada posición de referencia para realizar la medición de la posición del dispositivo Leap Motion, el cual fué de 250 ms. En el caso del escenario dinámico, las mediciones fueron tomadas sin ningún tiempo de espera, esto es, cuando los objetos se encontraban en movimiento. Se desarrolló una aplicación en Qt para llevar a cabo las mediciones, la cual permitió el manejo del módulo lineal a través del puerto serie, así como el tomar las mediciones del dispositivo Leap Motion y el cálculo de la precisión y repetibilidad. Como se mencionó anteriormente se siguieron las especificaciones del estándar ISO 9283, en el cual se menciona que la temperatura debe ser constante y no oscilar más de 2 grados. En este caso, la temperatura fué de aproximadamente 20°C, y se mantuvo una iluminación de aproximadamente 450 lux en el cuarto en donde se realizaron los experimentos. Se obtuvieron un total de 92,400 mediciones, 13,200 en el caso del lápiz, y 79,200 del modelo de la mano de plástico.

2.3 Resultados

En esta sección se presentan los resultados de las mediciones basado en el diseño de los experimentos descritos en la sección 2.2. El tiempo que se requirió para llevar a cabo las mediciones de ambos objetos de referencia en el escenario estático, fué de 7.6 hrs, mientras que el tiempo requerido para el caso dinámico fué de 6.7 hrs, dando un total de 14.3 hrs, lo cual se debió completamente a las características del motor del módulo lineal de alta precisión Rexroth.

Primero se presentan los resultados del modelo de la mano de plástico en los dos escenarios, seguido por los resultados de las mediciones del lápiz igualmente en los dos escenarios.

2.3.1. Mediciones del modelo de la mano de plástico

2.3.1.1. Escenario estático

En la tabla 2.1 se muestran los valores obtenidos en el escenario estático de la precisión y repetibilidad de las mediciones del modelo de la mano de plástico.

	Eje	x	y	z	media
Palma	Precisión (mm)	4.06	0.44	0.31	1.60
	Repetibilidad (mm)	2.67	4.13	2.71	3.17
Dedo pulgar	Precisión (mm)	2.62	1.38	0.70	1.57
	Repetibilidad (mm)	3.45	3.53	2.31	3.10
Dedo índice	Precisión (mm)	0.54	0.78	0.92	0.75
	Repetibilidad (mm)	1.95	1.46	2.25	1.89
Dedo medio	Precisión (mm)	0.67	0.65	0.59	0.64
	Repetibilidad (mm)	2.22	2.00	2.49	2.24
Dedo anular	Precisión (mm)	0.60	0.56	0.44	0.53
	Repetibilidad (mm)	3.08	1.54	2.23	2.29
Dedo meñique	Precisión (mm)	1.56	0.66	0.45	0.89
	Repetibilidad (mm)	3.31	1.40	1.81	2.17

Tabla 2.1: Precisión y repetibilidad en los tres ejes cartesianos

A continuación se muestran las gráficas de barras de la precisión y repetibilidad sobre los ejes x , y y z en condiciones estáticas. En la gráfica de la figura 2.5 se puede observar que la mayor precisión se logró sobre el eje z en la medición de la posición de la palma, el valor obtenido fué 0.31 mm. Además observamos que el valor menos preciso igual fue obtenido en las mediciones de la palma pero esta vez sobre el eje x , el valor que se obtuvo fué 4.06 mm. Se obtuvieron valores más consistentes sobre los tres ejes en las mediciones de los dedos índice, medio y anular.

Con respecto a la repetibilidad, en la gráfica de la figura 2.6, las mejores mediciones (el menor valor de todos, de 1.40) fue en el dedo meñique. Mientras que el peor valor se obtuvo en la medición de la palma sobre el eje y . Se observa que en las mediciones de los dedos índice y medio sobre los tres ejes se obtuvieron valores mejores así como más consistentes.

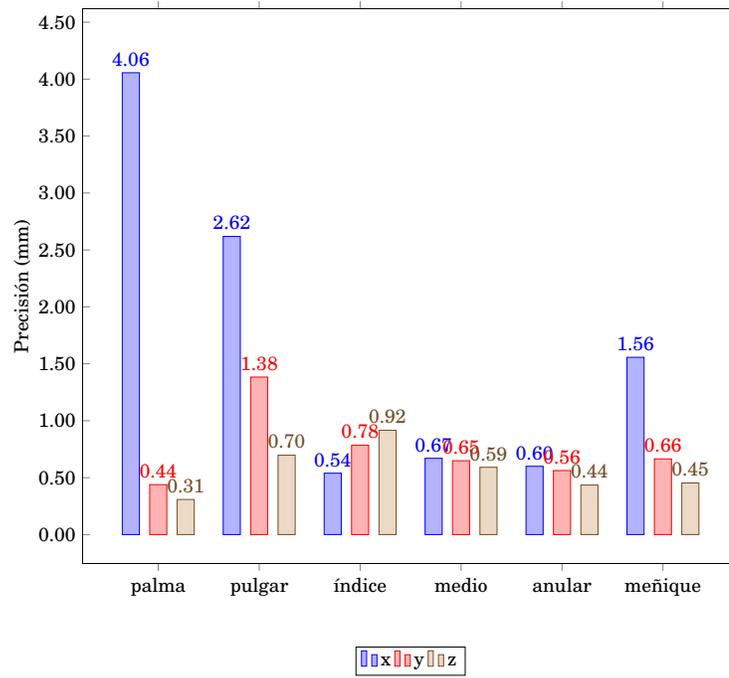


Figura 2.5: Gráfica de la precisión de la mano de plástico sobre los ejes x , y y z , en el escenario estático.

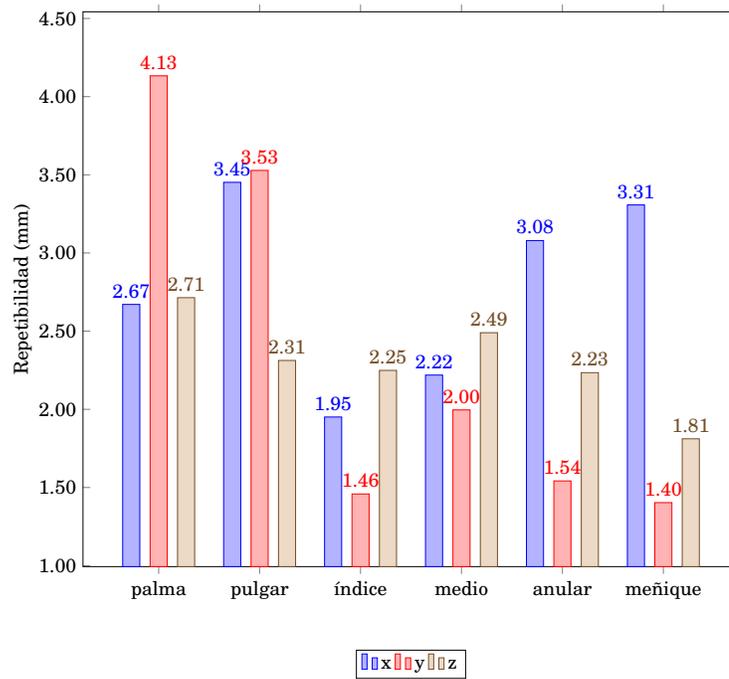


Figura 2.6: Gráfica de la repetibilidad de la mano de plástico sobre los ejes x , y y z , en el escenario estático.

2.3.1.2. Escenario dinámico

En la tabla 2.2 se muestran los valores obtenidos en el escenario dinámico de la precisión y repetibilidad de las mediciones del modelo de la mano de plástico.

	Eje	x	y	z	media
Palma	Precisión (mm)	1.09	7.32	1.75	3.39
	Repetibilidad (mm)	0.30	10.48	3.38	4.72
Dedo pulgar	Precisión (mm)	1.08	2.05	3.00	2.05
	Repetibilidad (mm)	0.42	7.01	2.85	3.42
Dedo índice	Precisión (mm)	0.41	2.83	2.06	1.77
	Repetibilidad (mm)	0.31	6.01	2.09	2.80
Dedo medio	Precisión (mm)	0.42	2.62	1.85	1.63
	Repetibilidad (mm)	0.27	6.82	2.67	3.25
Dedo anular	Precisión (mm)	0.56	2.14	1.74	1.48
	Repetibilidad (mm)	0.37	6.83	2.74	3.31
Dedo meñique	Precisión (mm)	1.56	1.66	1.82	1.68
	Repetibilidad (mm)	0.60	5.99	1.62	2.74

Tabla 2.2: Precisión y repetibilidad en los 3 ejes cartesianos

En la gráfica de la figura 2.7 se muestran los valores de la precisión de las mediciones de la mano de plástico en condiciones dinámicas. Se puede ver que la precisión más alta se obtuvo en las mediciones del pulgar sobre el eje x (0.41 mm). Y el peor valor se obtuvo en las mediciones de la palma sobre el eje y (7.32 mm). Además podemos notar que en general estos valores son menos precisos que en el caso del escenario estático.

En la gráfica de la figura 2.8 se muestran los valores de la repetibilidad de las mediciones de la mano plástico sobre los ejes x , y y z en condiciones dinámicas. Se puede observar que sobre el eje x se obtuvo la mejor repetibilidad, y sobre el eje y la peor. En comparación con los resultados de la repetibilidad en condiciones estáticas, los valores en condiciones dinámicas fueron mejores, aunque se hubiera podido pensar que esto sería todo lo contrario.

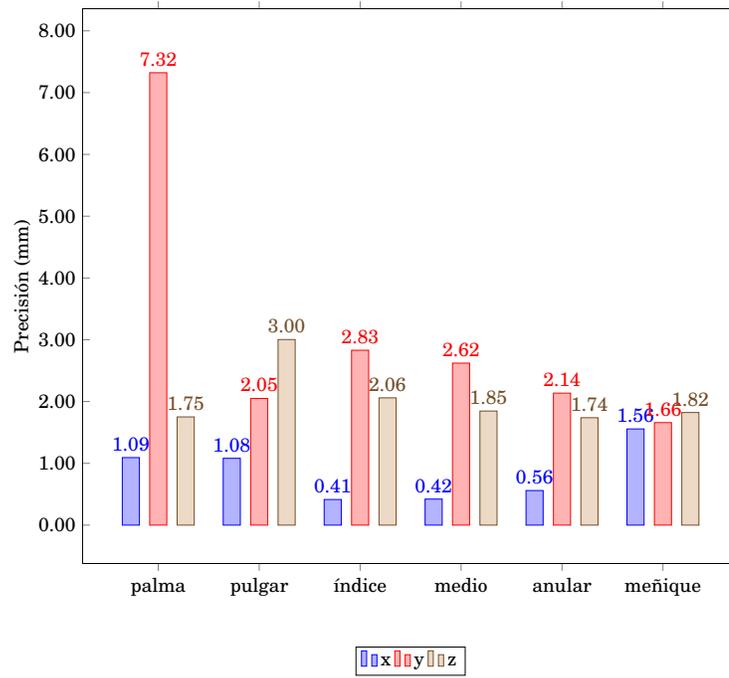


Figura 2.7: Gráfica de la precisión de la mano de plástico sobre los ejes x , y y z , en el escenario dinámico.

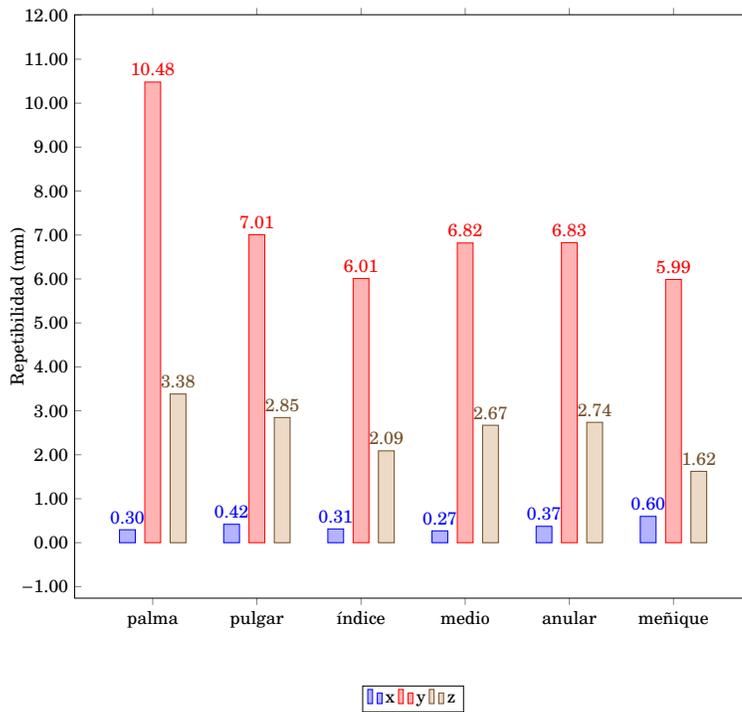


Figura 2.8: Gráfica de la repetibilidad de la mano de plástico sobre los ejes x , y y z , en el escenario dinámico.

2.3.2. Mediciones usando un lápiz

2.3.2.1. Escenario estático

En la tabla 2.3 se muestran los valores de la precisión y repetibilidad de las mediciones obtenidos en el escenario estático usando un lápiz.

Eje	x	y	z	media
Precisión (mm)	0.27	4.58	1.94	2.26
Repetibilidad (mm)	1.62	1.43	2.45	1.83

Tabla 2.3: Precisión y repetibilidad en los 3 ejes cartesianos usando un lápiz (escenario estático).

En la figura 2.9 se muestran las gráficas de la precisión y repetibilidad de las mediciones usando un lápiz en condiciones estáticas. Se observa que se obtuvo una muy buena precisión sobre el eje x , aunque una muy mala precisión sobre el eje y . Aunque por otro lado, la repetibilidad sobre el eje y fué mejor que la que se obtuvo sobre los otros dos ejes.

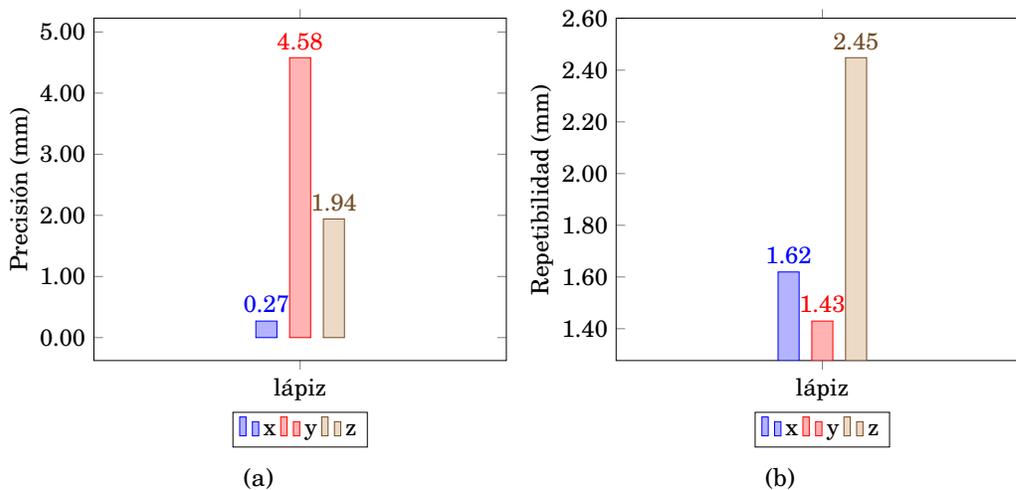


Figura 2.9: Precisión (a) y repetibilidad (b) del lápiz sobre los ejes x , y y z , en el escenario estático.

2.3.2.2. Escenario dinámico

En la tabla 2.4 se muestran los valores obtenidos en el escenario dinámico de la precisión y repetibilidad de las mediciones usando un lápiz.

En la figura 2.10 se muestran las gráficas de la precisión y repetibilidad obtenidos de las mediciones usando un lápiz en condiciones dinámicas.

Eje	x	y	z	media
Precisión (mm)	0.87	0.91	2.99	1.59
Repetibilidad (mm)	2.75	0.51	2.27	1.84

Tabla 2.4: Precisión y repetibilidad en los 3 ejes cartesianos

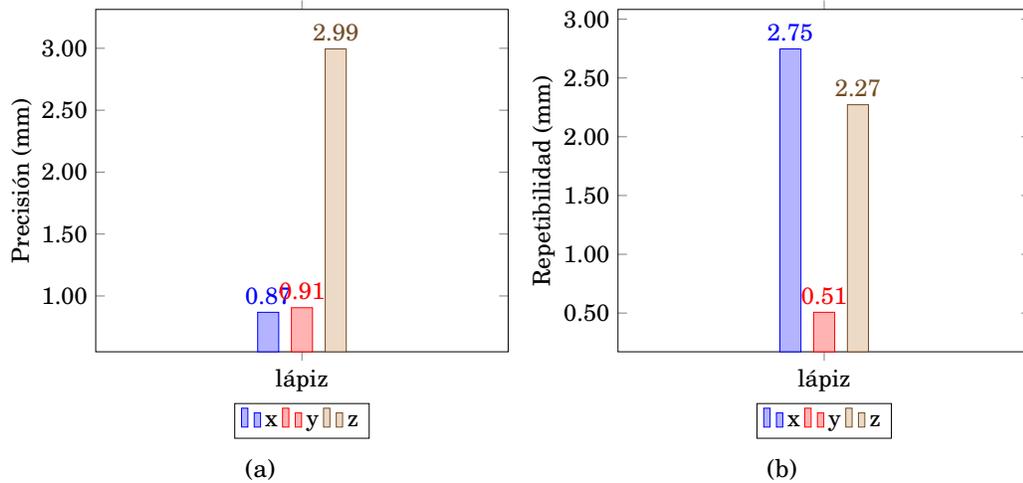


Figura 2.10: Precisión (a) y repetibilidad (b) del lápiz sobre los ejes x , y y z , en el escenario dinámico.

Se puede observar en la figura 2.10(a) que la precisión sobre el eje y fue mucho mejor comparada con la del escenario estático (figura 2.9(a)), así como también la repetibilidad fue mejor sobre este eje. La peor precisión se obtuvo en las mediciones tomadas sobre el eje z , y la peor repetibilidad se obtuvo sobre el eje x .

3

ANÁLISIS DEL DISPOSITIVO MYO

En este capítulo se presenta el análisis realizado al dispositivo Myo de Thalmic Labs. Este dispositivo es una banda que se coloca en el antebrazo. Se describirán a detalle los sensores que lo componen, así como los datos que proporcionan. También se explica como se utilizó su giróscopo incorporado para controlar el desplazamiento lineal, tanto en 2D como en 3D, en una aplicación. En la sección 3.1, se realiza la descripción del dispositivo Myo, se describe la manera en como funciona, como los sensores que lo componen. La sección 3.2 ofrece una descripción de los gestos que reconoce el dispositivo. En la sección 3.3, se describe la forma en que se utilizó el giróscopo para controlar un desplazamiento en 3D.

3.1 Descripción del dispositivo

El dispositivo Myo fué desarrollado por la compañía de origen canadiense Thalmic Labs. La versión final del dispositivo fué lanzada en el verano de 2014. Myo es un dispositivo cuya funcionalidad se basa en el control por gestos, el cual se coloca en alguno de los antebrazos como si fuera un brazalete y se comunica con otros dispositivos compatibles a través de la tecnología Bluetooth®Smart Wireless. El dispositivo está compuesto internamente por una unidad de medición inercial (UMI), 8 sensores electromiográficos y una conexión Bluetooth. Cuenta con un procesador ARM Cortex M4, además de leds indicadores y retroalimentación háptica (vibraciones cortas y largas). Contiene una batería de litio la cual es recargable a través del puerto Micro-USB, y su duración es de un día entero de uso. El UMI esta conformado por un magnetómetro, un giróscopo y un acelerómetro, de 3 ejes cada uno. Los cuales proporcionan datos espaciales, mismos que pueden ser obtenidos mediante el API de desarrollo del dispositivo. Los sensores electromiográficos, detectan las señales eléctricas de los músculos del brazo y proporcionan los valores a través del API de desarrollo. El

reconocimiento de los gestos que proporciona el dispositivo se hace utilizando los valores que proporcionan estos sensores.



Figura 3.1: Dispositivo Myo. Es una banda que se coloca alrededor del antebrazo.

Myo es compatible con los dispositivos que contengan adaptador USB para Bluetooth, OpenGL 2.1 o superior, así como los siguientes sistemas operativos:

- Windows 7, 8 y 10
- MAC OS X 10.8 (Mountain Lion) y versiones actuales
- IOS 7 y superior, iPad 3ra y 4ta generación; iPad Air, iPad Air 2; iPhone 4s, 5, 5c, 5s, 6, 6 Plus; iPod Touch 5ta generación, iPad Mini 1ra y 2da generación y iPad Mini 3
- Android 4.3 (Jelly Bean) y superior

3.2 Gestos que reconoce el dispositivo

El dispositivo proporciona el reconocimiento de 5 gestos:

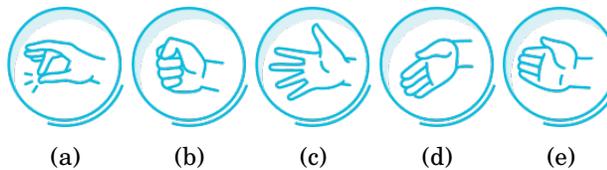


Figura 3.2: Gestos predeterminados que reconoce el dispositivo. (a) *tic doble*, (b) *puño*, (c) *dedos abiertos*, (d) *onda izquierda* y (e) *onda derecha*

Las señales eléctricas generadas por el antebrazo al realizar estos movimientos, son reconocidas por el software del dispositivo, el cual a su vez, identifica dichos gestos. Viendo la figura 3.2, el gesto *tic doble*, se logra al chocar rápidamente dos veces el dedo medio contra el pulgar. El gesto *puño*, se logra cerrando la mano. El gesto *dedos abiertos*, se logra al separar y extender todos los dedos simultáneamente. Los gestos *onda derecha* y *onda izquierda*, se logran al hacer movimientos con la muñeca hacia adentro y hacia afuera con los dedos juntos y extendidos.

3.3 Uso del giróscopo para controlar un desplazamiento en 3D

Como se mencionó en la sección 3.1, el dispositivo Myo cuenta con una UMI, la cual a su vez esta compuesta por un acelerómetro, un magnetómetro y un giróscopo, de tres ejes cada uno. Se utilizaron los valores arrojados por el giróscopo para el desplazamiento de un objeto sobre los ejes x , y y z . El fabricante proporciona código y explicación para lograr el control del cursor del ratón en [23] en 2D. Para lograr el desplazamiento de un objeto en 3D, se utilizó el código que el fabricante proporciona en [23], el cual se adaptó para poder utilizar el eje faltante. El código básicamente integra la velocidad angular dada por el giróscopo para generar tres desplazamientos sobre los ejes x , y y z . Las tareas principales de la implementación son:

- Filtrar los valores que entrega el giróscopo.
- Convertir los datos del giróscopo al marco de referencia del mundo.
- Integrar los datos del giróscopo para generar los valores x , y y z .
- Determinar la ganancia.

En las siguientes secciones se da una explicación más detallada de estas tareas.

3.3.1. Filtrar los valores que entrega el giróscopo

El filtro de la media móvil es el más común en el procesamiento digital de señales, debido a que es el filtro digital más fácil de entender y usar [24]. Este filtro tiene ventajas y desventajas, para nuestro propósito, se determinó que era viable el utilizarlo, ya que no se necesitaba un filtro muy complejo, además de que es óptimo (tiene un costo computacional muy bajo en comparación con otros filtros más complejos) en tareas simples y muy fácil de implementar. La fórmula utilizada se muestra a continuación:

$$y[i] = \frac{1}{M} \sum_{j=0}^{M-1} x[i+j] \quad (3.1)$$

donde $x[]$ es la señal de entrada, $y[]$ la señal de salida, y M es el número de orden o muestras utilizadas.

Para atenuar los valores del sensor y eliminar el ruido, se utilizó el filtro de la media móvil de orden 4, esto es, se obtuvo la media de los valores de las cuatro muestras previas. Esto se hace cada que ocurre el evento `onGyroscopeData` de la clase implementada del tipo `myo::DeviceListener`.

3.3.2. Convertir los datos del giróscopo al marco de referencia del mundo

El giróscopo nos proporciona valores de la velocidad angular en grados por segundo del movimiento del brazo, en su marco de referencia propio, esto es, el marco de referencia del sensor dentro del sistema de coordenadas del dispositivo [25]. Los datos filtrados del giróscopo se pasaron de grados a radianes antes de realizar la conversión del marco de referencia. Lo que se hizo fue rotar el marco de referencia del sensor a la orientación del marco de referencia del mundo para poder alinearlos con la pantalla. Para esto, se utilizó la función `myo::rotate()` que proporciona el API de desarrollo del dispositivo, la cual, recibe como parámetros un cuaternión y un vector con los valores en radianes del giróscopo.

En la figura 3.3 se muestra la configuración del sistema de coordenadas del dispositivo Myo.

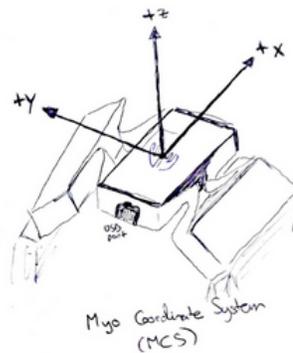


Figura 3.3: Sistema de coordenadas del dispositivo Myo.

3.3.3. Extraer los parámetros x , y y z del vector con los datos del giróscopo

Una vez que se obtuvo el vector con los valores del giróscopo en el marco de referencia del mundo, se multiplica por la duración de un marco (en segundos) y se obtienen los valores para los parámetros x , y y z . El parámetro x se obtiene de la componente z negativa del vector, el parámetro y de la componente y positiva del vector, y el parámetro z de la componente x positiva del vector. El código se muestra en el apéndice A, en la página 57.

3.3.4. Determinar la ganancia

Como se menciona en [23], la ganancia es la relación entre el movimiento del brazo y los píxeles. Esto es, el determinar la ganancia es para saber cual es el factor por el que tenemos que multiplicar x , y y z , para convertir de grados por segundo a píxeles. Algunos factores importantes que se toman en cuenta son la *sensibilidad* y la *aceleración*. La *sensibilidad* se refiere a que tan suave o fuerte tiene que ser el movimiento del brazo para poder lograr un desplazamiento del cursor. La *aceleración* es el factor que determina que tan rápido se mueve el cursor.

3.4 Experimentos

Debido a la falta de equipo especializado, no se pudieron plantear experimentos para el análisis de las características de los sensores que lo componen. El costo comercial de éste tipo de equipos de calibración, resulta más caro que el del mismo dispositivo Myo. Es por esto, que desafortunadamente no se pudo realizar un análisis de los sensores electromiográficos ni de los sensores que componen la UMI.

4

ANÁLISIS DEL DISPOSITIVO SPACE NAVIGATOR

En este capítulo se explica el análisis de la repetibilidad realizado al dispositivo Space Navigator y se presentan los resultados que se obtuvieron mediante tablas y gráficas. Este dispositivo no entrega valores de desplazamiento, por lo que no se pudieron realizar pruebas de precisión de estos valores como se hizo en el caso del dispositivo Leap Motion. En la sección 4.1, se describe el funcionamiento del dispositivo Space Navigator. La sección 4.2 muestra la implementación del manejador del dispositivo. En la sección 4.3, se describe el experimento realizado para el análisis de repetibilidad de los valores que proporciona el dispositivo sobre los ejes x y z . La sección 4.4 muestra los resultados que se obtuvieron del experimento realizado en forma de tablas y gráficas.

4.1 Descripción del dispositivo

El dispositivo Space Navigator es un ratón 3D desarrollado por la compañía norteamericana 3Dconnexion [26], el cual proporciona seis grados de libertad (6 GdL), esto significa que se tienen tres movimientos de traslación y tres movimientos de rotación, en cada uno de los ejes cartesianos x , y y z . Este dispositivo está diseñado principalmente para interactuar con aplicaciones de diseño asistido por computadora (CAD, por sus siglas en inglés). Este dispositivo es compatible con los sistemas operativos convencionales como Windows, MAC OS X y GNU/Linux, utiliza el puerto USB para el envío de los datos que detectan sus sensores. Está diseñado para el control de objetos, así como para el manejo de la vista de la cámara en aplicaciones 3D. Para esto, el dispositivo proporciona valores de la velocidad de traslación y/o rotación del objeto o vista que el usuario está tratando de controlar. Además cuenta con dos botones, uno en cada costado. A diferencia de los ratones comunes que se desplazan físicamente, este dispositivo permanece fijo, lo que se mueve es la parte superior de él, y se utiliza aplicando presiones y torsiones en su parte superior. El modo de uso de

este dispositivo puede asemejarse al de una palanca o *joystick*, la cual se puede presionar, jalar o inclinar para obtener una respuesta de parte de los sensores que lo componen.



Figura 4.1: Dispositivo Space Navigator.

4.2 Implementación del manejador del dispositivo

Uno de los objetivos de la presente tesis, es la creación del manejador del dispositivo Space Navigator. Esto debido a que el manejador que proporciona el fabricante es *cerrado*, es decir, no se puede modificar, ni se puede acceder directamente a los valores que los sensores arrojan. El tener acceso a los valores que los sensores proporcionan, es vital para poder realizar el análisis de la repetibilidad de los mismos. Es por esto, que en esta sección se presenta la implementación que se realizó del manejador del dispositivo para los sistemas operativos GNU/Linux y Mac OS X.

4.2.1. Tipos de manejadores en los sistemas Unix

Cuando se desarrollan manejadores en sistemas Unix, se tienen dos tipos:

1. *Manejadores en espacio del núcleo.*
2. *Manejadores en espacio del usuario.*

Los manejadores normalmente se ejecutan en el *espacio del núcleo*, ya que el manejo de interrupciones y el mapeo de los recursos de hardware requieren de privilegios que sólo el espacio del núcleo esta autorizado a proporcionar. Sin embargo, esto presenta algunas desventajas como:

1. *Sobrecarga en las llamadas al sistema.* Cada llamada al núcleo requiere del cambio de modo de usuario a modo de supervisor o administrador, y después cambiar a modo de usuario de nuevo. Lo cual toma tiempo y puede convertirse en un cuello de botella si las llamadas son muy frecuentes.
2. *Curva de aprendizaje.* Hay mucho que aprender para poder desarrollar un manejador en el espacio del núcleo.
3. *Estabilidad de la interfaz.* El API del espacio del núcleo es menos estable que el API del espacio del usuario.

4. *Difícil de depurar.*

5. *Errores son más fatales.*

Los manejadores en el *espacio del usuario* no están exentos de tener problemas e inconvenientes. Un desafío que se presenta en el desarrollo de un manejador en el espacio del usuario, es el *manejo de las interrupciones*. Debido a que el manejo de interrupciones requiere de privilegios superiores a los del modo usuario, resulta imposible implementar un manejador de interrupciones en el espacio del usuario. Sin embargo, se tienen dos opciones para lidiar con este problema: la primera es no utilizar interrupciones, y la segunda es tener un manejador pequeño en el espacio del núcleo únicamente para manejar la interrupción. El no utilizar interrupciones significa que se tienen que estar leyendo constantemente los valores del dispositivo para obtener los datos.

El manejador del dispositivo Space Navigator que se implementó es un manejador en el *espacio del usuario* y no utiliza interrupciones.

4.2.2. Comunicación dispositivo-computadora

Como se mencionó en la sección 4.1 página 25, la forma en la que el dispositivo Space Navigator envía los datos es a través del puerto USB. La interfaz USB es una solución apropiada cuando se desea utilizar una computadora para comunicarse con un dispositivo externo, debido a que es fácil de usar, la transferencia de datos es rápida y confiable, bajo costo, bajo consumo de energía, entre otras [27].

La arquitectura USB define varias clases de especificaciones donde define los formatos para los datos que serán transmitidos, además del número y tipo de los criterios de evaluación requeridos y opcionales para los dispositivos de la clase. Dentro de estas clases se encuentra la clase de *Dispositivos de Interfaz Humana (DIH)*. La clase DIH consiste principalmente de dispositivos que los seres humanos utilizan para controlar una computadora como: teclados, ratones, palancas de mando o controles de juegos de video, impresoras, entre otros [27].

Para la implementación de la comunicación del manejador con el dispositivo a través del puerto USB de una computadora, se utilizó la biblioteca multiplataforma HIDAPI, la cual sirve de interfaz entre una aplicación y dispositivos de la clase DIH [28]. Esta biblioteca es más útil con dispositivos personalizados (definidos por el proveedor) de la clase DIH, como el Space Navigator.

4.2.2.1. Abrir conexión con el dispositivo

Una vez que se definió que se utilizaría la biblioteca HIDAPI, lo siguiente que se tiene que hacer es utilizar la función *hid_open()* para abrir la comunicación con el dispositivo. La función recibe como parámetros: el identificador del vendedor (*vendor_id*), el identificador del producto (*product_id*), y el número de serie (*serial_number*). La función regresa

un apuntador al objeto *hid_device* en caso de ser exitosa la conexión o nulo en caso contrario.

La declaración de la función *hid_open()* se muestra a continuación:

```
1 HID_API_EXPORT hid_device* HID_API_CALL hid_open(unsigned short vendor_id, ↵  
2 unsigned short product_id, const wchar_t *serial_number  
)
```

Se define el modo de no-bloqueo con la función *hid_set_nonblocking()*, la cual recibe como parámetros: el apuntador al objeto *hid_device* y el entero 1 para activar el modo no-bloqueo o 0 para desactivarlo.

La declaración de la función *hid_set_nonblocking()* se muestra a continuación:

```
1 int HID_API_EXPORT HID_API_CALL hid_set_nonblocking(hid_device *device, int ↵  
nonblock)
```

4.2.3. Lectura de los valores que entregan los sensores

Como se mencionó en la sección 4.2.1 página 26, el manejador implementado es del tipo de los manejadores en el *espacio del usuario* y como no hace uso de interrupciones, lee constantemente los valores del dispositivo. Para realizar la lectura de los valores que proporciona el dispositivo, se utiliza la función *hid_read()* de la biblioteca HIDAPI, la cual recibe como parámetros: el apuntador al objeto *hid_device*, una lista de caracteres que sirve para almacenar los valores leídos, y el tamaño de la lista. La función regresa el número de bytes leídos o -1 en caso de error.

La declaración de la función *hid_read()* se muestra a continuación:

```
1  
2 int HID_API_EXPORT HID_API_CALL hid_read(hid_device *device, unsigned char *data↵  
, size_t length)
```

4.2.4. Interpretación de los valores entregados por los sensores

Los valores que se obtuvieron de la función *hid_read()* se interpretan de la siguiente manera:

1. *El número de bytes leídos es 3.* Los datos provienen de uno de los dos botones del dispositivo. Si el valor de la lista de caracteres en la posición 1 es igual a 01, entonces se presionó el botón izquierdo, de lo contrario si es igual a 02, se presionó el botón derecho.

2. *El número de bytes leídos es 7.* Los datos son valores de la velocidad de traslación o rotación. Si el valor de la lista de caracteres en la posición 0 es igual a 01, entonces son valores de la velocidad de traslación, en caso de ser igual a 02, se trata de valores de la velocidad de rotación.

4.2.4.1. Valores de la velocidad de traslación y rotación

Cuando el número de bytes leídos es igual a 7, quiere decir que el tamaño de la lista de caracteres es de 7. El primer byte determina el tipo de movimiento, 01 para traslación y 02 para rotación. Tres bytes para los valores de la velocidad de traslación o rotación sobre los ejes x , y y z , y van de 0 a 255. Y tres bytes de desbordamiento sobre los ejes x , y y z .

1er byte	2do byte	3er byte	4to byte	5to byte	6to byte	7mo byte
Tipo de mov.	Valor en eje x	Desbord. en eje x	Valor en eje y	Desbord. en eje y	Valor en eje z	Desbord. en eje z

Tabla 4.1: Valores de los bytes de la lista de caracteres.

Los valores de desbordamiento nos dan la posibilidad de obtener valores mayores a 255, así como también de obtener valores negativos. El significado de los valores de los bytes de desbordamiento se muestran en la tabla 4.2.

Valor	Significado
00	Valor positivo. No hay desbordamiento
01	Valor positivo. Hay desbordamiento
255	Valor negativo. No hay desbordamiento
254	Valor negativo. Hay desbordamiento

Tabla 4.2: Valores y significado de los bytes de desbordamiento.

En la implementación del manejador del dispositivo, se definió una constante igual a 400 para obtener el valor +/- 400 cuando hay desbordamiento. Esto es, cuando el valor del byte de desbordamiento es igual a 01, el valor es de +400, cuando el valor del byte de desbordamiento es 254, el valor es de -400. Cuando no hay desbordamiento los valores se encuentran en el rango -255 a 255.

El código se muestra en el apéndice B en la página 61.

4.3 Descripción del experimento

Para la realización del experimento se ocuparon, un dispositivo Space Navigator, un módulo lineal de alta precisión, un circuito electrónico que sirve para controlar el motor a pasos del módulo lineal, un modelo de una mano de plástico fijado a una base de madera en forma de escuadra, una aplicación desarrollada en Qt y C++ para la automatización de las mediciones y una computadora portátil. Además, como parte de los objetivos particulares de la presente tesis, se desarrolló un manejador del dispositivo Space Navigator para poder acceder a los valores en crudo de los sensores que lo componen, mismo que será descrito más adelante. El módulo lineal de alta precisión *Bosch Rexroth AG D-97419*, como ya se mencionó en la sección 2.2 en la pág. 8, proporciona un solo grado de libertad (1 GdL) y esta conformado por un motor a pasos bipolar, el cual tiene una precisión $50 \mu\text{m}$ (micrómetros) por cada paso. Debido a las limitantes del módulo lineal, así como a las características del dispositivo Space Navigator, no fue posible realizar mediciones sobre el eje y , por lo que solamente se tomaron mediciones sobre los ejes x y z en ambos sentidos (positivo y negativo).

4.3.1. Configuración de las mediciones

Para la realización de las mediciones sobre los ejes x y z , se acopló el modelo de la mano de plástico a una base de madera en forma de escuadra, esta se colocó firmemente en el módulo lineal de alta precisión. El dispositivo Space Navigator fue fijado fuertemente a una mesa, lo cual evitó que el dispositivo se desplazara al momento de realizar las mediciones. Se desarrolló una aplicación de software utilizando Qt y C++ para automatizar el movimiento del módulo lineal y para realizar la captura de las mediciones de los valores que arrojan los sensores del dispositivo. Los resultados de las mediciones obtenidas se guardaron en archivos de texto plano para su posterior análisis. Se utilizó una computadora portátil Macbook Pro con procesador Intel®Core™i7 CPU @ 2.9 GHz con 8 GB de RAM. En la figura 4.2 se muestra gráficamente la conformación de dicho experimento.



Figura 4.2: Configuración del experimento para el análisis del dispositivo Space Navigator.

4.3.2. Experimentos

Para la evaluación de la estabilidad de los valores proporcionados por los sensores del dispositivo, se tomaron mediciones de seis posiciones de referencia $p = \{0.6 \text{ mm}, 1.2 \text{ mm}, 1.8 \text{ mm}, 2.4 \text{ mm}, 3.0 \text{ mm}, 3.6 \text{ mm}\}$, las cuales se realizaron sobre los ejes x y z en ambos sentidos (positivo y negativo) y el número n de repeticiones realizadas fue 30. El procedimiento llevado a cabo para la ejecución de las mediciones fue el siguiente, estando el modelo de la mano de plástico lo más cerca al dispositivo sin llegar a tocarlo, se desplazó el módulo lineal a la primera posición de referencia (0.6 mm), el tiempo de espera para tomar la medición de los valores arrojados por los sensores fue de dos segundos, posteriormente se avanzó a la siguiente posición (1.2 mm) y se repitió el procedimiento anterior, hasta llegar a la posición final (3.6 mm). Como se mencionó anteriormente, se realizaron 30 repeticiones de cada una de las mediciones. La aplicación desarrollada en Qt y C++ ayudó a automatizar el procedimiento descrito anteriormente. Se validó perfectamente que el modelo de la mano de plástico al inicio de cada una de las repeticiones de las mediciones estuviera lo más cerca posible del dispositivo pero sin llegar a tocarlo para evitar que se afectará la medición.

4.3.3. Análisis estadístico

Debido a las limitantes del experimento no se pudo definir un sistema metrológico como en el caso del análisis del dispositivo Leap Motion descrito en la sección 2.2.2, por lo que se optó por la realización de un análisis estadístico.

El análisis estadístico fue la herramienta utilizada para realizar el análisis de los datos obtenidos de las mediciones realizadas del experimento descrito en la sección 4.3, en particular las medidas empleadas fueron la media aritmética y la desviación estándar.

4.4 Resultados

En esta sección se presentan los resultados de las mediciones obtenidas de los experimentos definidos en la sección 4.3. Se obtuvieron un total de 720 mediciones, 360 por cada eje cartesiano en los dos sentidos (180 en el positivo y 180 en el negativo). El tiempo requerido para la realización completa del experimento fue de aproximadamente 4 hrs.

Se hallaron los valores mínimo y máximo en cada una de las mediciones de cada posición de referencia definida en el conjunto p , además se presenta el cálculo de la media y de la desviación estándar. En las siguientes tablas se muestran estos valores tomados sobre los ejes x y z en ambos sentidos (positivo y negativo).

En la tabla 4.3 se puede observar que en el eje x en el sentido positivo, conforme aumenta la distancia desplazada, los valores de la desviación estándar son cada vez mayores, siendo el peor valor de 16.49.

Desplazamiento (mm)	Valor mínimo	Valor máximo	Media aritmética	Desviación estándar
0.6	4.0	8.0	6.83	0.82
1.2	15.0	30.0	18.70	4.39
1.8	26.0	60.0	33.73	8.80
2.4	41.0	81.0	55.80	13.07
3.0	59.0	114.0	76.83	16.96
3.6	78.0	123.0	98.20	16.49

Tabla 4.3: Valores obtenidos de las mediciones sobre el eje x positivo.

Desplazamiento (mm)	Valor mínimo	Valor máximo	Media aritmética	Desviación estándar
0.6	-10.0	-6.0	-6.97	0.98
1.2	-42.0	-32.0	-34.80	1.85
1.8	-68.0	-43.0	-55.87	7.42
2.4	-92.0	-60.0	-77.97	7.55
3.0	-124.0	-73.0	-108.07	10.51
3.6	-128.0	-112.0	-126.33	3.91

Tabla 4.4: Valores obtenidos de las mediciones sobre el eje x negativo.

En la tabla 4.4 se observa que los valores de la desviación estándar mejoraron con respecto a los de la tabla 4.3, en este caso, los valores no empeoran conforme la distancia aumenta. El peor valor en esta ocasión fue 10.50.

Desplazamiento (mm)	Valor mínimo	Valor máximo	Media aritmética	Desviación estándar
0.6	6.0	9.0	6.80	0.75
1.2	36.0	44.0	40.70	1.51
1.8	56.0	91.0	72.70	7.31
2.4	91.0	116.0	106.80	8.23
3.0	112.0	125.0	121.33	2.75
3.6	129.0	130.0	129.97	0.18

Tabla 4.5: Valores obtenidos de las mediciones sobre el eje z positivo.

Los valores de la desviación estándar calculada en el eje z positivo son mejores que en los obtenidos sobre el eje x en ambos sentidos, esto se muestra en la tabla 4.5

Desplazamiento (mm)	Valor mínimo	Valor máximo	Media aritmética	Desviación estándar
0.6	-7.0	-4.0	-4.70	0.74
1.2	-36.0	-26.0	-28.00	1.73
1.8	-71.0	-60.0	-62.30	2.60
2.4	-102.0	-76.0	-92.20	4.37
3.0	-120.0	-92.0	-114.13	4.31
3.6	-123.0	-120.0	-120.62	1.00

Tabla 4.6: Valores obtenidos de las mediciones sobre el eje z negativo.

En la tabla 4.6, se observa que los valores obtenidos en el eje z negativo fueron mejores, ya que estuvieron más cerca a la media que los de las tablas 4.3, 4.4 y 4.5.

La gráfica de la desviación estándar de los valores sobre el eje x y eje z se muestra en la figura 4.3.

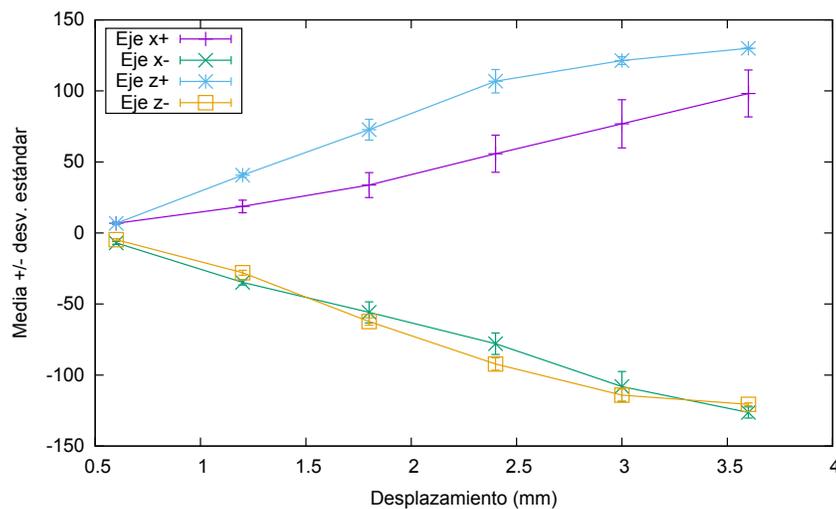


Figura 4.3: Gráfica de la media +/- y de la desviación estándar de los valores obtenidos del experimento.

En la gráfica de la figura 4.3 se observa que los valores fueron mejores en las mediciones sobre el eje z , especialmente en el sentido negativo. Los peores valores se obtuvieron sobre el eje x en el sentido positivo, porque tienen la mayor desviación estándar, llegando a ser de 16.49 el peor valor. Con esto podemos interpretar que la repetibilidad de los sensores no están buena como se pudiera esperar, y no son estables en los valores que arrojan en los ejes x y z .

5

USO DE LOS DISPOSITIVOS EN UNA APLICACIÓN PARA EVALUAR SU DESEMPEÑO

En el presente capítulo se presenta el estudio para el análisis del desempeño de los dispositivos Leap Motion, Myo y Space Navigator bajo el marco de la Ley de Fitts en aplicaciones en 2D y 3D. El primer experimento que se presenta, es el de evaluar la acción de *selección* en una aplicación de software en 2D, en donde se le pide a los usuarios que seleccionen un rectángulo en color verde, el cual, es seleccionado aleatoriamente al inicio de cada condición. Este experimento está basado en el esquema de *tarea en serie* definido en [29]. Para el segundo experimento, se planteó un juego en 3D, en donde se tienen dos objetos 3D, el primero de ellos es desplazado a una distancia D en la dirección P y la meta es llevar el segundo objeto a la posición del primero. El estándar ISO 9241-9 presenta una metodología basada en la Ley de Fitts [30]. Para el presente estudio, sólo se tomaron algunas consideraciones de ese estándar.

La selección de objetos (dar click) es una tarea fundamental en las interfaces de usuario convencionales. La acción de *selección* en interfaces de usuario en 2D, ha recibido una gran atención y ha sido bien modelado por la Ley de Fitts [31], la cual puede ser utilizada para:

- Modelar la manera en que los usuarios desempeñan tareas de selección
- Medir el desempeño del usuario en el uso de la aplicación
- Comparar el desempeño del usuario entre varios dispositivos de entrada

La Ley de Fitts también puede ser aplicada para tareas de selección en aplicaciones 3D, las medidas más comunes son la velocidad y la precisión. En [32] Pino *et al.* hacen un análisis del dispositivo Kinect, comparado con el ratón convencional. En el 2015, Bachmann *et al.* en [17] realizaron el estudio del dispositivo Leap Motion, comparándolo con los dispositivos de entrada convencionales como el ratón y el teclado.

El presente estudio utiliza el marco de la Ley de Fitts para hacer el análisis de los dispositivos: Leap Motion, Myo y Space Navigator, en dos aplicaciones, una en 2D y la otra en 3D.

En la sección 5.1, se describe la Ley de Fitts, así como el número de participantes en el experimento y el equipo utilizado. En la sección 5.2, se describe la aplicación en 2D realizada, así como los resultados obtenidos. En la sección 5.3, se describe la aplicación en 3D realizada, así como los resultados que se obtuvieron.

5.1 Metodología

La Ley de Fitts es el marco de trabajo teórico más utilizado para describir y comparar el desempeño de diferentes dispositivos de entrada en aplicaciones de software. Una de las tareas que resuelve este marco de trabajo, es la de predecir el tiempo que toma el mover el cursor del ratón a un botón y darle click. Otra de las tareas que resuelve, es la de estimar el rendimiento como un índice de desempeño del sistema motor humano.

El desempeño de los dispositivos en las aplicaciones diseñadas en el presente estudio es evaluado en conformidad con la Ley de Fitts [31]. Para lo cual, se tuvo que definir la dificultad de la tarea y fue cuantificada como el índice de dificultad (I_d en bits) utilizando la teoría de la información. Se utilizó la formulación de Shannon [33] la cual es la versión aceptada ampliamente y recomendada en [29] para calcular el índice de dificultad:

$$I_d = \log_2 \left(\frac{D}{A} + 1 \right) \quad (5.1)$$

donde D es la distancia entre el punto de origen y el objeto establecido como objetivo, y A es el ancho del objetivo. Las unidades del I_d son bits y surgen de la base 2 del logaritmo. El I_d es utilizado como el factor para manipular la dificultad de la tarea. Como la variable dependiente, el tiempo de movimiento (t_m) fue utilizado para evaluar el desempeño de la tarea. Según la Ley de Fitts el t_m es una función lineal del I_d , caracterizando un movimiento a lo largo de una simple dimensión:

$$t_m = a + bI_d \quad (5.2)$$

donde a es el tiempo de reacción y b es el tiempo para transmitir un bit, ambas son constantes y son determinadas con una regresión lineal de varias mediciones del índice de dificultad.

Fitts también propuso cuantificar la tasa humana del procesamiento de la información en los movimientos dirigidos utilizando “bits por segundo” como unidades. Fitts lo llamó “índice de desempeño”, al día de hoy es mejor conocido como rendimiento (r). La fórmula recomendada en [29] para el cálculo del r es la siguiente.

$$r = \frac{I_{de}}{t_m} \quad (5.3)$$

I_{de} es usado como el “ajuste para la precisión” e involucra el cálculo primero del ancho efectivo del objetivo (A_e) como:

$$A_e = 4.133 \sigma_x \quad (5.4)$$

donde σ_x es la desviación estándar observada en las selecciones de cada una de las repeticiones de un usuario en cada intento. En esencia, calculado de esta manera se captura lo que el participante en realidad hizo, en lugar de lo que lo pusieron a hacer. El ajuste necesario para esto queda de la siguiente manera:

$$I_{de} = \log_2 \left(\frac{D}{A_e} + 1 \right) \quad (5.5)$$

5.1.1. Participantes

El número de participantes en la evaluación de la aplicación en 2D fue de seis (seis hombres), de los cuales uno mencionó ser zurdo. Y para la aplicación en 3D, el número de participantes fue de cinco (cuatro hombres y una mujer), de los cuales, un hombre es zurdo. Todos con un promedio de edad de 31 años y acostumbrados a utilizar interfaces de usuario modernas. Ninguno tenía problemas de salud (visión, presión, etc). Cabe mencionar que ninguno de los participantes tenía experiencia previa en el uso de los dispositivos Leap Motion, Myo y Space Navigator.

5.1.2. Equipo utilizado

Para la implementación de las aplicaciones en 2D y 3D para la evaluación de la Ley de Fitts, se requirió de los dispositivos Leap Motion, Myo y Space Navigator. Además del framework Qt y de una laptop Macbook pro Intel® Core™ i7 CPU @ 2.9GHz con 8 GB de RAM con sistema operativo Mac OS X.

5.2 Descripción de la aplicación en 2D

La aplicación consiste de una ventana en donde se dibujan dos rectángulos de ancho A separados por una distancia D . Aleatoriamente se selecciona uno de los rectángulos y se pinta de color verde, el otro se pinta de color negro. Con alguno de los dispositivos se mueve el cursor con el que se tiene que seleccionar el rectángulo verde. Al seleccionarse, esto es, al mover el cursor sobre el rectángulo verde, y aplicar alguna acción, se pinta el rectángulo de negro. Este proceso se repite 15 veces, donde el rectángulo verde se intercambia con el

5. USO DE LOS DISPOSITIVOS EN UNA APLICACIÓN PARA EVALUAR SU DESEMPEÑO

negro en cada repetición.

Son siete distintas condiciones, definidas a distintas distancias y ancho del rectángulo y se muestran en la tabla 5.1.

Condición	Distancia del objetivo, D	Ancho del objetivo, A	Índice de dificultad, I_d
1	8.00	1.30	2.80
2	14.00	1.30	3.60
3	16.00	1.00	4.10
4	12.00	0.70	4.20
5	22.00	1.00	4.50
6	18.00	0.70	4.70
7	20.00	0.50	5.40

Tabla 5.1: Condiciones del experimento para la evaluación de la Ley de Fitts en la aplicación 2D

Las unidades de la tabla 5.1 son unidades arbitrarias de la aplicación en OpenGL, para transformarlas a pixeles se usan las relaciones:

$$\frac{x - \text{mín}_x}{\text{máx}_x - \text{mín}_x} = \frac{u}{l}, \quad y \quad \frac{y - \text{mín}_y}{\text{máx}_y - \text{mín}_y} = \frac{v}{a} \quad (5.6)$$

donde (x, y) es el punto en unidades arbitrarias, (u, v) es el punto en pixeles, a es el ancho de la ventana en pixeles, donde se dibujan los rectángulos; l es el largo de la ventana en pixeles. El área de dibujo esta especificada por dos puntos mín y máx.

En la figura 5.1 se muestran las pantallas de las siete condiciones usadas en la prueba 2D.

5.2.1. Forma de usar la aplicación 2D

Primero se tiene que seleccionar el dispositivo que se va a usar. Presionando la tecla L , M o S , significa seleccionar el dispositivo L =Leap Motion, M =Myo o S =Space Navigator.

La tarea que se realiza y evalúa es desplazar el cursor y seleccionar el rectángulo objetivo. La forma en que se realiza cambia según el dispositivo:

1. Para el Leap Motion el desplazamiento se logra manteniendo el gesto *un dedo* y desplazando la mano dentro del espacio físico de detección del dispositivo. La selección se logra con el gesto *dos dedos*.
2. Para el Myo el desplazamiento se hace moviendo el antebrazo. La selección es a través del gesto *puño*.
3. Para el Space Navigator con la mano izquierda se presiona para lograr el desplazamiento del cursor. La selección se logra presionando el botón izquierdo del dispositivo.

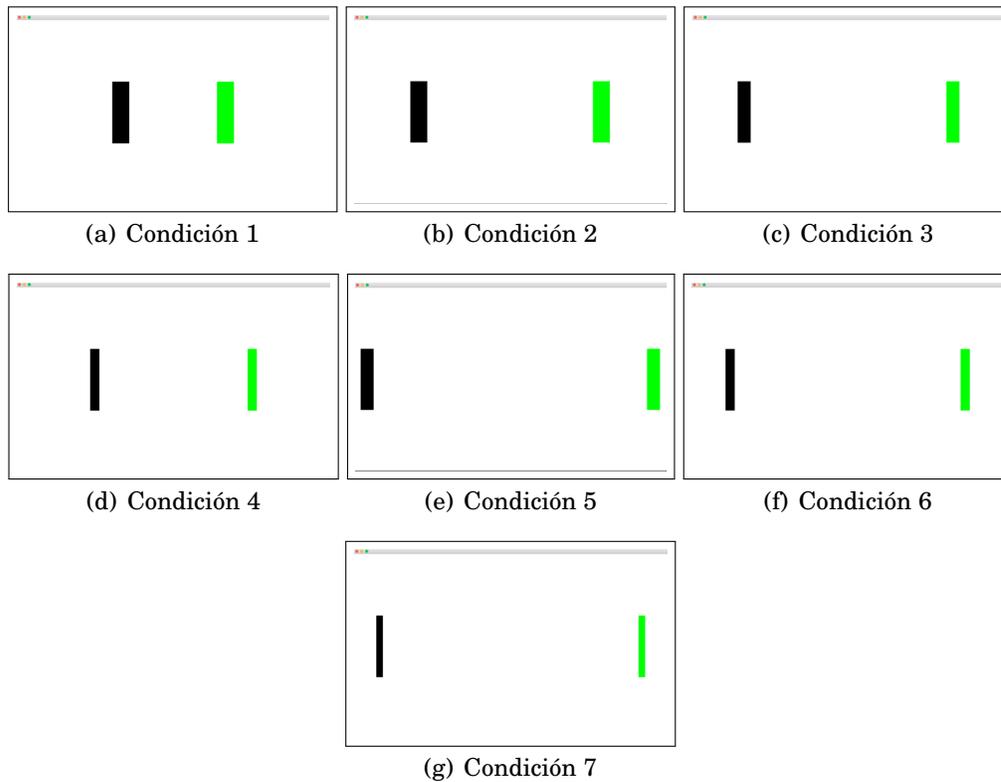


Figura 5.1: Las siete condiciones de la aplicación en 2D.

La forma general de utilizar la aplicación es de la siguiente manera:

- Al inicio el cursor se sitúa a la mitad del rectángulo negro.
- Se mide el tiempo desde que se mueve el cursor 0.05 unidades arbitrarias, hasta que se realiza la selección.
- Seleccionar fuera del rectángulo verde se contabiliza como fallo.
- Se realizaron 15 repeticiones bajo la misma condición, esto es, misma distancia D y mismo ancho A .

A los seis usuarios al final se les pidió que contestaran un cuestionario para medir la usabilidad de los mismos.

5.2.2. Resultados

En esta sección, se presentan los resultados obtenidos del desempeño de los tres dispositivos en la aplicación en 2D. Estos resultados están basados en el marco de trabajo de la Ley de Fitts descrito en la sección 5.1 en la página 36. Se planteó un experimento basado en la Ley de Fitts, para el análisis de la tarea de selección con los tres dispositivos objetos de estudio de esta tesis. Se presentan resultados del tiempo promedio que tarda un usuario en

5. USO DE LOS DISPOSITIVOS EN UNA APLICACIÓN PARA EVALUAR SU DESEMPEÑO

desplazar el cursor hacía un objeto marcado como objetivo y seleccionarlo, del rendimiento, de la tasa de error al seleccionar un objeto, así como los resultados de la usabilidad.

En la sección 5.2.2.1 se dan los resultados del tiempo del movimiento, en la sección 5.2.2.2 los resultados del rendimiento, en la sección 5.2.2.3 la tasa de error y en la sección 5.2.2.4.

5.2.2.1. Tiempo del movimiento promedio

La media del tiempo de movimiento (t_m) en todo el experimento fue de 2.9 s. La media de t_m con el dispositivo Leap Motion fue de 2.61 s, la del dispositivo Myo de 2.44 s, y la del dispositivo Space Navigator de 3.76 s. El dispositivo Myo obtuvo los valores más bajos, mientras que el dispositivo Space Navigator los más altos. Se nota que al incrementar el número de I_d el tiempo aumentó también, con excepción del número I_d 5 en los dispositivos Myo y Space Navigator, ya que el valor en este número de I_d fue más alto que en el número de I_d 7.

I_d	2.80	3.60	4.10	4.20	4.50	4.70	5.40
Leap Motion (s)	2.0	2.0	2.64	2.61	2.98	2.46	3.56
Myo (s)	2.18	2.27	2.46	2.23	2.77	2.49	2.72
Space Navigator (s)	3.24	3.22	3.55	3.57	4.62	3.86	4.23

Tabla 5.2: Media de los tiempos de movimiento (t_m) de los dispositivos por cada índice de dificultad (I_d).

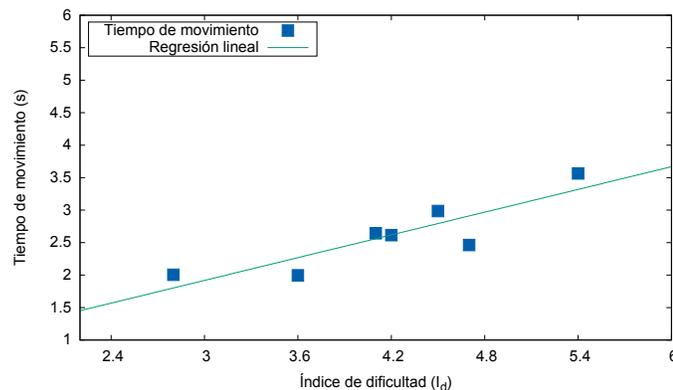


Figura 5.2: Gráfica de las medias de los tiempos de movimiento (t_m) del dispositivo Leap Motion.

Con respecto a los valores obtenidos de las constantes a y b de la fórmula 5.2, estos se muestran en la tabla 5.3. Podemos observar que el tiempo para transmitir un bit es mejor con el dispositivo Myo (0.22 s), y el tiempo de reacción mejor lo proporciona el dispositivo

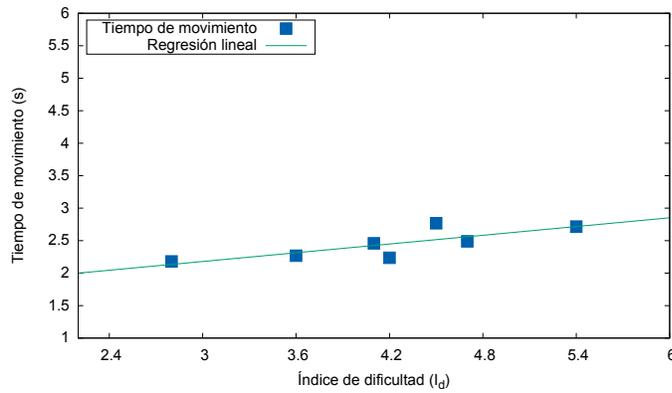


Figura 5.3: Gráfica de las medias de los tiempos de movimiento (t_m) del dispositivo Myo.

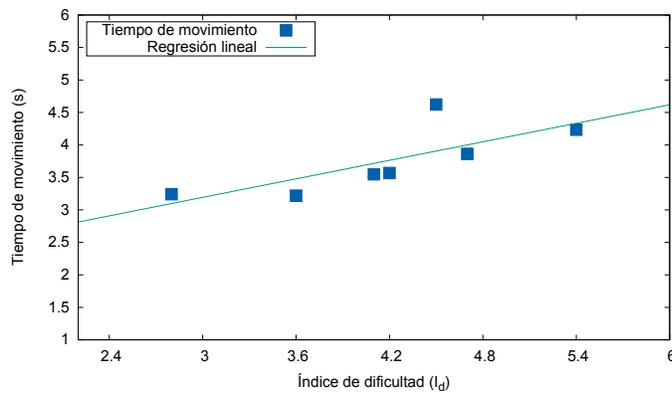


Figura 5.4: Gráfica de las medias de los tiempos de movimiento (t_m) del dispositivo Space Navigator.

Leap Motion (0.17 s). El tiempo de reacción típico al utilizar el ratón convencional es de 0.2 a 0.3 segundos [34], un tiempo muy corto, en comparación con los obtenidos de los dispositivos Myo y Space Navigator, aunque el dispositivo Leap Motion tuvo un mejor tiempo (0.17 s). En este experimento se tienen dos tiempos de reacción, el primero es al comenzar a mover el cursor, y el segundo es al seleccionar el objetivo (dar click).

Constantes	a (s)	Error	b (s/bit)	Error
Leap Motion	0.17	+/- 0.61 (361.8%)	0.58	+/- 0.14 (24.5%)
Myo	1.51	+/- 0.32 (21.5%)	0.22	+/- 0.08 (33.94%)
Space Navigator	1.77	+/- 0.78 (44.31%)	0.47	+/- 0.18 (38.77%)

Tabla 5.3: Valores de las constantes a y b para cada dispositivo, así como su error estándar asintótico, calculados con regresión lineal con gnuplot.

5.2.2.2. Rendimiento

El promedio total del rendimiento (r) durante todo el experimento fue de 1.13 bits/s. La media del rendimiento del dispositivo Leap Motion fue de 1.2 bits/s, la del dispositivo Myo de 1.4 bits/s y la del dispositivo Space Navigator de 0.78 bits/s. El rendimiento de los dispositivos por cada índice de dificultad (I_d) se presenta en la tabla 5.4 y la figura 5.5.

I_d	2.80	3.60	4.10	4.20	4.50	4.70	5.40	Media
Leap Motion (bits/s)	1.17	1.42	1.14	1.14	1.09	1.40	1.04	1.2
Myo (bits/s)	1.18	1.36	1.39	1.53	1.31	1.48	1.51	1.4
Space Navigator (bits/s)	0.63	0.80	0.84	0.84	0.70	0.90	0.87	0.78
Media total (%)								1.13

Tabla 5.4: Media del rendimiento (r) de los dispositivos por cada índice de dificultad (I_d).

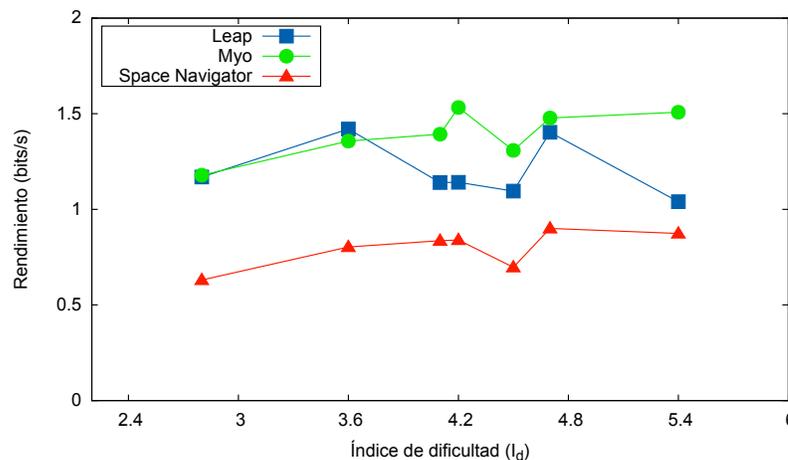


Figura 5.5: Gráfica de la media del rendimiento (r) de los dispositivos por cada índice de dificultad (I_d).

5.2.2.3. Tasa de error

El promedio de la tasa de error en todo el experimento con los tres dispositivos fue de 5.6%. La tasa de error al utilizar el dispositivo Leap Motion fue de 8.8%, la del dispositivo Space Navigator de 6.5% y la del dispositivo Myo de sólo 1.1%. Al interpretar estos resultados podemos decir que con el dispositivo Myo fue más fácil el realizar las tareas definidas, en contraparte, al utilizar el dispositivo Leap Motion se presentaron mayores problemas para llevarlas a cabo. En la tabla 5.5 y en la figura 5.6, se muestra la tasa de error de cada dispositivo por índice de dificultad (I_d). Para el dispositivo Leap Motion se nota que al ser

más grande el I_d la tasa de error aumentó, para el dispositivo Myo se mantuvo constante y muy baja para toda la amplitud de índices de dificultad; el dispositivo Space Navigator muestra una tasa de error mayor para $I_d < 3.6$.

I_d	2.80	3.60	4.10	4.20	4.50	4.70	5.40	Media
Leap Motion (%)	4.3	3.2	9.1	3.2	8.2	11.8	19.6	8.8
Myo (%)	0.0	2.2	2.2	0.0	2.2	0.0	1.1	1.1
Space Navigator (%)	10.0	9.1	4.3	6.3	5.3	6.3	4.3	6.5
Media total (%)								5.6

Tabla 5.5: Tasa de error de los dispositivos por cada índice de dificultad (I_d).

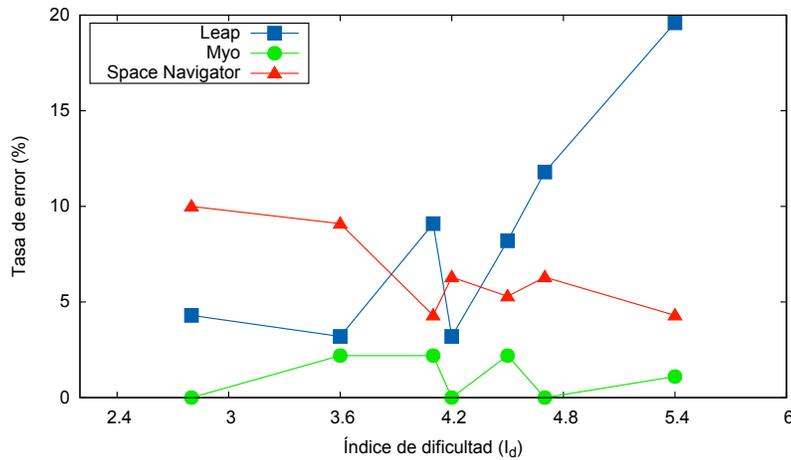


Figura 5.6: Gráfica de la tasa de error de los dispositivos por cada índice de dificultad (I_d).

5.2.2.4. Usabilidad

Para medir la usabilidad de los dispositivos, se diseñó un cuestionario basado en las recomendaciones del ISO-9241-9. Las preguntas realizadas a los usuarios al terminar de usar cada uno de los dispositivos fueron las siguientes:

1. Esfuerzo mental requerido.
2. Facilidad para desplazar el cursor.
3. Fátiga de la muñeca.
4. Fátiga del hombro.
5. Fátiga del brazo.
6. Facilidad para apuntar con el cursor el objeto.
7. Facilidad para seleccionar el objeto.
8. Facilidad de interacción en general.
9. Este dispositivo puede reemplazar al ratón convencional (no = 1, si = 5)

Se les pidió dar valores entre 1 (bajo) y 5 (alto) a cada una de las preguntas anteriores. La media de cada una de las respuestas de los seis participantes se muestran en la tabla 5.6.

Dispositivos	Leap Motion	Myo	Space Navigator
Pregunta 1	3.3	1.5	4.2
Pregunta 2	4.5	4.8	2.0
Pregunta 3	1.5	1.0	1.2
Pregunta 4	4.2	1.0	1.0
Pregunta 5	5.0	1.0	1.0
Pregunta 6	3.3	4.5	1.2
Pregunta 7	2.5	4.5	2.0
Pregunta 8	3.0	4.7	2.7
Pregunta 9	1.0	4.3	1.0

Tabla 5.6: Valores de las medias de las respuestas al cuestionario realizado a los seis participantes.

En la tabla 5.7 se agrupan los resultados de las medias del tiempo de movimiento (t_m), del rendimiento (r) y de la tasa de error, de los dispositivos en la aplicación en 2D. En esta tabla se ve claramente que el dispositivo Myo tuvo los mejores valores en cada uno de los aspectos analizados.

Dispositivo	Tiempo de movimiento (s)	Rendimiento (bits/s)	Tasa de error (%)
Leap Motion	2.61	1.2	8.8
Myo	2.44	1.4	1.1
Space Navigator	3.76	0.78	6.5
Medias totales (%)	2.9	1.13	5.6

Tabla 5.7: Medias del tiempo de movimiento (t_m), rendimiento (r) y tasa de error de los dispositivos en la aplicación en 2D.

5.3 Descripción de la aplicación en 3D

La aplicación consiste de una escena 3D en donde se dibujan dos teteras, una de color verde, y la otra de color azul, las cuales están superpuestas al inicio de cada condición. Se realiza una acción con alguno de los dispositivos para desplazar la tetera de color verde a una distancia D en una dirección P . Después se usa alguno de los dispositivos para desplazar la tetera de color azul, la cual se tiene que superponer a la tetera de color verde. Al superponerse, esto es, al desplazar la tetera de color azul hasta la posición casi exacta de la tetera de color verde, la tetera de color verde se pinta de rojo y se tiene un tiempo de espera de 3 segundos. Después de esto, la tetera de color rojo se pinta de verde nuevamente y la tetera de color azul se desplaza a la posición inicial. Este proceso se repite 10 veces durante cinco condiciones definidas para el experimento.

Las cinco condiciones definidas a distintas distancias para el experimento se muestran en la tabla 5.8:

Condición	Distancia del objetivo, D	Ancho del objetivo, A	Índice de dificultad, I_d
1	7.00	3.00	3.00
2	8.00	3.00	3.17
3	9.00	3.00	3.32
4	10.00	3.00	3.46
5	11.00	3.00	3.58

Tabla 5.8: Condiciones del experimento para la evaluación de la Ley de Fitts en la aplicación 3D

Al igual que en la aplicación en 2D, las unidades de la tabla 5.8 son unidades arbitrarias de la aplicación en OpenGL, y para transformarlas a pixeles se usan las relaciones 5.6 de la página 38 para obtener los valores en pixeles de las componentes u y v , y la siguiente relación para la componente w del punto en pixeles (u, v, w):

$$\frac{z - \text{mín}_z}{\text{máx}_z - \text{mín}_z} = \frac{w}{p} \quad (5.7)$$

donde z es la componente del punto en unidades arbitrarias, w la componente del punto en pixeles, p es la profundidad de la ventana en pixeles.

5.3.1. Forma de usar la aplicación 3D

Primero se tiene que seleccionar el dispositivo que se va a usar. Presionando la tecla L , M o S , significa seleccionar el dispositivo L =Leap Motion, M =Myo o S =Space Navigator.

La tarea que se realiza y evalúa es desplazar la tetera de color azul y superponerla a la tetera de color verde. La forma en que se realiza cambia según el dispositivo:

1. Para el Leap Motion el desplazamiento al inicio de cada condición de la tetera de color verde a la distancia D en la dirección P , se logra haciendo el gesto *dos dedos*. La tetera de color verde se desplaza automáticamente hasta su posición final, una vez que se detuvo se realiza el gesto *dos dedos* nuevamente, para habilitar el desplazamiento de la tetera de color azul. El desplazamiento de la tetera de color azul se realiza manteniendo el gesto *un dedo* y desplazando la mano dentro del espacio físico de detección del dispositivo.
2. Para el Myo el desplazamiento al inicio de cada condición de la tetera de color verde a la distancia D en la dirección P , se logra haciendo el gesto *dedos abiertos*. La tetera de color verde se desplaza automáticamente hasta su posición final, una vez que se detuvo se realiza el gesto *dedos abiertos* nuevamente, para habilitar el desplazamiento de la tetera de color azul. El desplazamiento de la tetera de color azul se realiza moviendo el antebrazo.
3. Para el Space Navigator el desplazamiento al inicio de cada condición de la tetera de color verde a la distancia D en la dirección P , se logra presionando el botón izquierdo del dispositivo. El desplazamiento de la tetera de color azul se realiza presionando con la mano izquierda el dispositivo.

La forma general de utilizar la aplicación es de la siguiente manera:

- Al inicio de cada condición las dos teteras se encuentran superpuestas en la posición inicial.
- Una vez que la tetera de color verde se desplazó a su posición final, el usuario puede mover la tetera de color azul.
- Se mide el tiempo desde que se desplaza la tetera de color azul 0.05 unidades arbitrarias, hasta que se desplaza hasta la posición casi exacta de la tetera de color verde, esto es, al superponer ambas teteras.
- Se realizaron 10 repeticiones bajo la misma condición, esto es, misma distancia D y mismo dirección P .

Al final del experimento, se les pidió a los usuarios que contestaran un cuestionario, para analizar la usabilidad de los dispositivos utilizados.

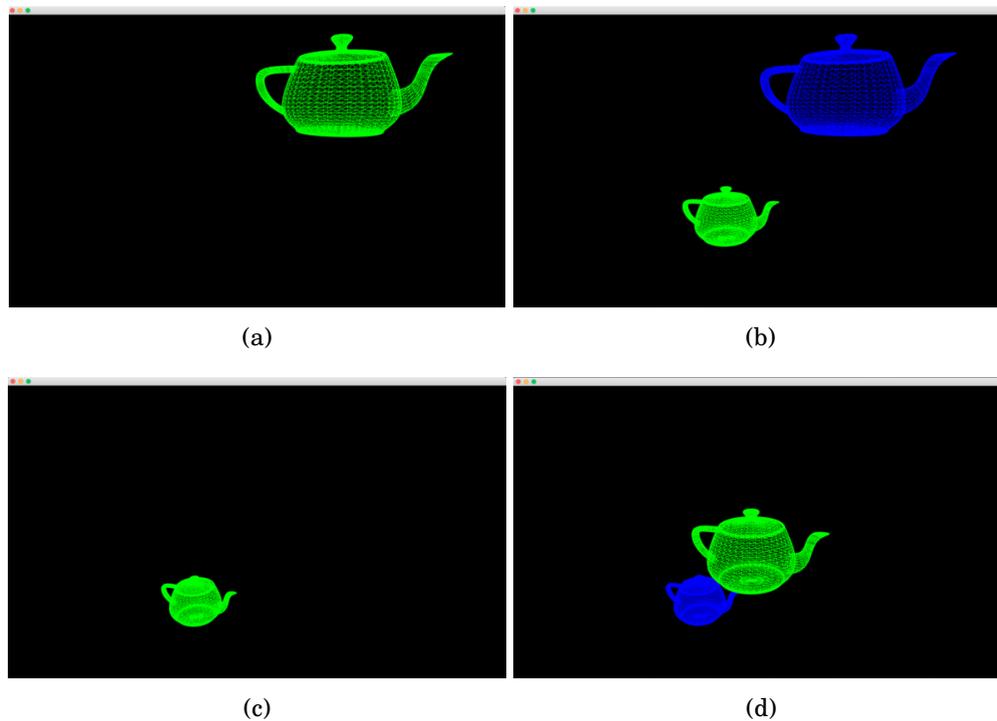


Figura 5.7: (a) Posición inicial, las dos teteras están superpuestas, (b) La tetera de color verde (*objetivo*), es desplazada una distancia D en la dirección 2 (hacia la esquina inferior izquierda). (c) Posición inicial, misma situación que en (a), (d) La tetera de color verde (*objetivo*), es desplazada una distancia D en la dirección 1 (hacia la esquina superior derecha)

5.3.2. Resultados

En esta sección, se presentan los resultados obtenidos del desempeño de los tres dispositivos en la aplicación en 3D. Estos resultados están basados en el marco de trabajo de la Ley de Fitts descrito en la sección 5.1 en la página 36. Se planteó un experimento para el análisis del desplazamiento de objetos en una aplicación en 3D, esto con los tres dispositivos objetos de estudio de esta tesis. En esta aplicación sólo se evaluó el tiempo de movimiento promedio que tarda un usuario en desplazar un objeto hacia una distancia D en una dirección P logrando superponerlo a otro objeto en el espacio 3D, esto debido a que en el ISO 9241-9 no hay una especificación para realizar el análisis de la Ley de Fitts en ambientes en 3D.

5.3.2.1. Tiempo del movimiento

La media del tiempo de movimiento (t_m) observada durante todo el experimento fue de 3.55 s. La media del tiempo de movimiento en la dirección 1 registrada fue de 3.76 s, mientras que la obtenida en la dirección 2 fue de 3.33 s. En las tablas 5.9 y 5.10 y en la figura 5.9 se muestran las medias de los tiempos de movimiento durante cada índice de dificultad (I_d). Se observa que la media del tiempo de movimiento al desplazar el objeto

5. USO DE LOS DISPOSITIVOS EN UNA APLICACIÓN PARA EVALUAR SU DESEMPEÑO

hacia la dirección 2 fue menor que el obtenido en la dirección 1. En las tablas también podemos observar que los mejores valores se obtuvieron realizando el desplazamiento del objeto con el dispositivo Leap Motion, y los peores al realizarlo con el dispositivo Space Navigator.

I_d	3.00	3.17	3.32	3.46	3.58	Media
Leap Motion (s)	2.34	2.34	2.56	3.12	3.33	2.74
Myo (s)	5.78	3.05	2.89	2.82	2.49	3.41
Space Navigator (s)	5.56	4.89	4.71	4.95	5.61	5.14
Media total (%)						3.76

Tabla 5.9: Media del tiempo de movimiento (t_m) de los dispositivos por cada índice de dificultad (I_d) en la dirección 1 en la aplicación en 3D.

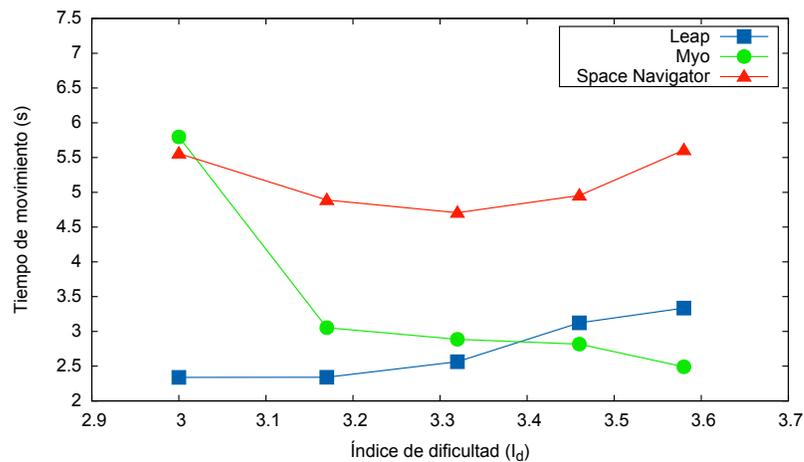


Figura 5.8: Gráfica de las medias de los tiempos de movimiento (t_m) de los dispositivos por cada índice de dificultad (I_d) en la dirección 1 en la aplicación en 3D.

I_d	3.00	3.17	3.32	3.46	3.58	Media
Leap Motion (s)	2.34	2.21	2.16	2.53	2.50	2.35
Myo (s)	3.56	4.22	2.91	2.87	2.97	3.30
Space Navigator (s)	4.33	4.19	4.68	3.99	4.53	4.35
Media total (%)						3.33

Tabla 5.10: Media del tiempo de movimiento (t_m) de los dispositivos por cada índice de dificultad (I_d) en la dirección 2.

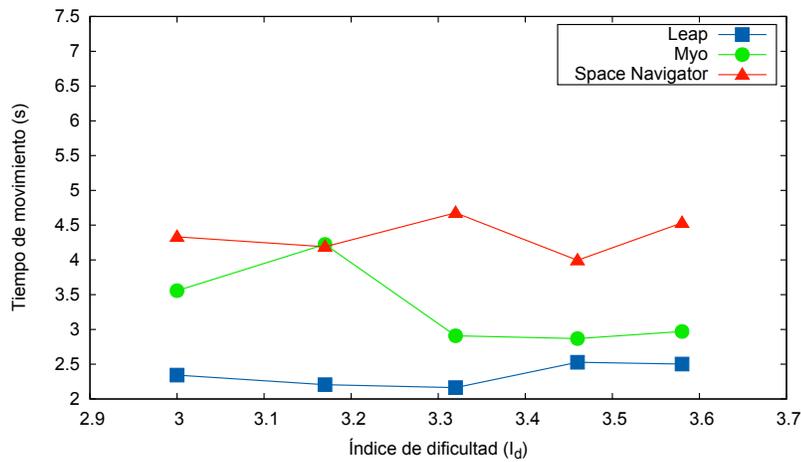


Figura 5.9: Gráfica de las medias de los tiempos de movimiento (t_m) de los dispositivos por cada índice de dificultad (I_d) en la dirección 2 en la aplicación en 3D.

5.3.2.2. Usabilidad

De igual forma que en la aplicación en 2D, para medir la usabilidad de los dispositivos, se aplicó un cuestionario basado en las recomendaciones del ISO-9241-9. Las preguntas realizadas a los usuarios al terminar de usar cada uno de los dispositivos fueron las siguientes:

1. Esfuerzo mental requerido.
2. Facilidad para desplazar el objeto.
3. Fátiga de la muñeca.
4. Fátiga del hombro.
5. Fátiga del brazo.
6. Facilidad para superponer los dos objetos.
7. Facilidad de interacción en general.

Se les pidió dar valores entre 1 (bajo) y 5 (alto) a cada una de las preguntas anteriores. La media de cada una de las respuestas de los cinco participantes se muestran en la tabla 5.11.

Dispositivos	Leap Motion	Myo	Space Navigator
Pregunta 1	1.2	1.4	2.0
Pregunta 2	4.8	3.6	4.6
Pregunta 3	1.0	1.0	1.0
Pregunta 4	4.4	1.2	1.0
Pregunta 5	5.0	1.2	1.0
Pregunta 6	4.2	3.0	3.6
Pregunta 7	4.6	3.6	4.0

Tabla 5.11: Valores de las medias de las respuestas al cuestionario realizado a los cinco participantes.

Dispositivo	Tiempo de movimiento (s)
Leap Motion	2.74
Myo	3.41
Space Navigator	5.14
Media total (%)	3.76

Tabla 5.12: Medias del tiempo de movimiento (t_m) de los dispositivos en la aplicación en 3D.

6

CONCLUSIONES

En esta tesis se realizó el análisis de los dispositivos Leap Motion, Myo y Space Navigator. Se diseñaron experimentos a la medida para cada uno de los dispositivos. Se evaluaron características como la precisión y repetibilidad de los sensores que componen cada uno de los dispositivos. Además se realizaron dos aplicaciones, una en 2D y la otra en 3D, en las que se evaluó el desempeño de los dispositivos en la interacción con estas aplicaciones utilizando el marco teórico de la Ley de Fitts. En la sección 6.1, se dan las conclusiones del dispositivo Leap Motion. En la sección 6.2, se dan las conclusiones del dispositivo Myo. En la sección 6.3, se dan las conclusiones del dispositivo Space Navigator. En la sección 6.4, se dan las conclusiones finales de los tres dispositivos estudiados y analizados en la presente tesis. Finalmente, en la sección 6.5, se muestra el trabajo futuro.

6.1 Leap Motion

Se analizó el dispositivo Leap Motion, utilizando como sistema de referencia un módulo lineal de alta precisión *Bosch Rexroth AG D-97419* con una base en la cual se acoplaron alternadamente dos objetos como referencia, un modelo de una mano de plástico y un lápiz. Las principales características analizadas fueron la precisión y la repetibilidad. El análisis se realizó en base al estándar ISO 9283, con una modificación en el número de ciclos, en este análisis se utilizaron 200 ciclos en lugar de 30, debido a las conclusiones de Jeswiet y Helferty en [22], en la que mencionan que no son suficientes los 30 ciclos establecidos por el estándar para poder determinar de manera adecuada la repetibilidad. Primeramente se realizaron las mediciones del modelo de la mano de plástico, tanto en el escenario estático, como en el dinámico. Se realizaron las mediciones de las posiciones de los 5 dedos y del centro de la palma de la mano. Debido a que el sistema de referencia utilizado para realizar las mediciones sólo cuenta con un grado de libertad, se tuvieron que analizar los tres ejes

cartesianos por separado. En el momento de estar realizando las mediciones, se tuvo que reiniciar todo el procedimiento debido varios problemas con el dispositivo Leap Motion y su software. Uno de esos problemas fué que el servicio (*demonio* en linux) del dispositivo Leap Motion se detenía abruptamente, dejando de detectar los objetos. Otro problema fué que al estar monitoreando la aplicación *Visualizer* el objeto desaparecía, esto es, se perdía continuamente la detección y seguimiento de los objetos. Esto hizo que el tiempo para realizar las mediciones creciera en gran medida. Una vez que se tuvieron los resultados de la precisión y repetibilidad de los dos objetos de referencia, se pudo observar una gran inconsistencia en los valores. La mejor precisión se obtuvo de las mediciones del lápiz sobre el eje x en condiciones estáticas, la cual fué de 0.27 mm, por otro lado, la peor precisión se obtuvo en las mediciones del centro de la palma en condiciones dinámicas, cuyo valor fué 7.32 mm. En cuestión de repetibilidad, los mejores valores se obtuvieron al realizar las mediciones de la mano de plástico sobre el eje x en el escenario dinámico. No hubo una tendencia hacia ninguno de los escenarios, se obtuvieron buenos y malos resultados en ambos. En contraste con los resultados de Weichert *et al.* en [1], en donde obtuvieron una precisión de las posiciones por debajo de los 0.2 mm en condiciones estáticas y de 1.2 mm en condiciones dinámicas, los valores que se hallaron en el presente análisis para la precisión van desde los 0.27 mm hasta los 7.32 mm.

Por otro lado, para el análisis del desempeño del dispositivo Leap Motion en la interacción con aplicaciones en 2D y 3D, se utilizó el marco teórico de la Ley de Fitts y algunas recomendaciones del estándar ISO 9241-9. En la aplicación en 2D lo que se evaluó fue, la tasa de error, el tiempo de movimiento, el rendimiento y la usabilidad. Los resultados obtenidos muestran que la tasa de error fue de 8.8%, la más alta de los tres dispositivos. Esto quiere decir que para tareas de selección de objetos en 2D, el dispositivo Leap Motion no es el más adecuado. En cuanto al tiempo de movimiento, la media fue de 2.61 segundos y el tiempo de reacción de 1.76 segundos, lo cual nos indica que es rápido realizar el desplazamiento del cursor con este dispositivo. La media de su rendimiento fue de 1.2 bits por segundo. En cuanto a los resultados obtenidos del cuestionario de usabilidad, tuvo valores muy altos en cuanto a la fatiga del brazo y hombro y valores muy bajos en cuanto a la facilidad para seleccionar objetos. Debido a que no hay una recomendación en el estándar ISO 9241-9 de como realizar la evaluación en aplicaciones en 3D, la configuración del experimento fue diseñada en base a algunos estudios vistos de otros dispositivos [32]. En la aplicación en 3D se evaluó únicamente el tiempo de movimiento. Los resultados obtenidos muestran que es rápido el realizar el desplazamiento de objetos en 3D con este dispositivo.

Con estos resultados, podemos recomendar el no utilizarlo en aplicaciones donde se tengan que seleccionar objetos pequeños, así como evitar el usarlo por periodos de tiempo prolongados ya que causa mucha fatiga en el brazo.

6.2 Myo

Desafortunadamente no se pudo realizar un análisis de las características de los sensores que componen el dispositivo Myo. Esto debido a la falta de equipo especializado para

realizar el análisis de acelerómetros, giróscopos, magnetómetros y sensores electromiográficos. Se gastó mucho tiempo en la investigación de métodos alternos para realizar un análisis. Para el caso de los acelerómetros y giróscopos, se encontró un método en [35]. En vez de este método, se utilizó un filtro simple como se explicó en la sección 3.3.1 en la página 21.

Se realizó el análisis en la interacción con aplicaciones en 2D y 3D, utilizando el marco teórico de la Ley de Fitts. En el caso de la aplicación en 2D, se evaluaron la tasa de error, el tiempo de movimiento, el rendimiento y la usabilidad. La tasa de error del dispositivo a través de todo el experimento fue de 1.1%, la más baja de todos los dispositivos. La media del tiempo de movimiento fue de 2.44 segundos, el tiempo de reacción de 2.11 segundos y el tiempo para transmitir un bit de 0.08% (el mejor de los tres dispositivos). En cuanto al rendimiento, el valor fue el más alto (1.4 bits por segundo). Los resultados obtenidos del cuestionario de usabilidad, muestran que tuvo valores muy altos en cuanto a facilidad para apuntar y seleccionar objetos y muy bajos en cuanto la fatiga que produce en el brazo o muñeca. Con respecto al análisis realizado en la interacción en la aplicación en 3D, el tiempo de movimiento fue el segundo mejor detrás del dispositivo Leap Motion. Los valores del cuestionario muestran que el realizar el desplazamiento del objeto no les pareció una tarea fácil a los participantes.

Con base en los resultados del análisis realizado al dispositivo, podemos ver que su utilización en aplicaciones donde se tengan que seleccionar objetos pequeños es recomendada. Además, el dispositivo no produce fatiga alguna en los usuarios, por lo que se puede utilizar en periodos prolongados de tiempo. En cuanto a las aplicaciones en 3D, debido a que se tiene que integrar la velocidad angular para obtener el desplazamiento (ver sección 3.3 en la página 21), para desplazar el eje z es necesario rotar el antebrazo en si mismo, y es un movimiento antinatural para realizar el desplazamiento de objetos, por lo que no se recomendaría en aplicaciones para desplazamientos en 3D.

6.3 Space Navigator

Implementar el manejador del dispositivo en el espacio del usuario, resultó una buena elección para la presente tesis, ya que su desempeño fue aceptable en las aplicaciones desarrolladas. Además de que el tiempo de desarrollo fue relativamente corto. El manejador se integró bien con el framework Qt y su paradigma orientado a eventos, lo que permitió que se leyeran constantemente los valores de los sensores de una manera correcta. La frecuencia de muestreo máxima del dispositivo para que se obtengan valores correctos es de 60 Hz, por lo que se puede decir que lo hace en tiempo real.

Se realizó el análisis del dispositivo Space Navigator, utilizando el mismo marco de referencia descrito en la sección 6.1. En este análisis se evaluó la repetibilidad de los valores que proporcionan los sensores del dispositivo. Debido a las características físicas del dispositivo, sólo se pudieron analizar los valores sobre los ejes x y z en los sentidos positivo y negativo. Los valores arrojados por el experimento muestran que sobre el eje x la

repetibilidad fue mucho peor que los obtenidos sobre el eje z . Los valores arrojados por los sensores no son tan estables cuando se aplica la misma entrada.

En el análisis realizado del desempeño del dispositivo en la interacción en aplicaciones en 2D y 3D, se utilizó el marco teórico de la Ley de Fitts. En la aplicación en 2D se evaluó la tasa de error, el tiempo de movimiento, el rendimiento y la usabilidad. La tasa de error reportada fue de 6.5%. La media del tiempo de movimiento fue de 3.76 segundos y el tiempo de reacción de 2.99 segundos, el más alto de los tres dispositivos. La media de su rendimiento fue de 0.78 bits por segundo, el valor más bajo de los dispositivos. En cuanto a los resultados obtenidos del cuestionario de usabilidad, tuvo valores muy altos en cuanto al esfuerzo mental requerido y muy bajos en cuanto a la facilidad para desplazar el cursor. En la aplicación en 3D se evaluó únicamente el tiempo de movimiento. El tiempo de movimiento fue el más alto, por lo que, realizar el desplazamiento de objetos en 3D con este dispositivo es lento en comparación con los otros dos dispositivos. El cuestionario de usabilidad arrojó valores muy altos en lo que respecta a la facilidad para desplazar el objeto y muy bajos en cuanto a la fatiga que produce al utilizarlo.

Con base en los resultados de los experimentos, podemos ver que no resulta adecuada su utilización en aplicaciones en 2D. En aplicaciones en 3D es más recomendable utilizar el dispositivo.

6.4 Conclusiones finales

En la tabla 6.1 se muestra un comparación de los tres dispositivos en diferentes rubros.

Dispositivos	Leap Motion	Myo	Space Navigator
Frecuencia de muestreo (Hz).	0-200	0-60	0-60
Precisión (mm).	0.27-7.32	N/A	N/A
Repetibilidad (mm).	0.27-10.48	N/A	0.18-16.96
Facilidad de desplazamiento en 2D.	2	1	3
Facilidad de desplazamiento en 3D.	1	3	2
Facilidad de selección.	3	1	2
Fatiga del brazo.	3	2	1
Rapidez de desplazamiento en 2D.	2	1	3
Rapidez de desplazamiento en 3D.	1	2	3
Error en la selección.	3	1	2
Número de gestos predeterminados.	4	5	N/A

Tabla 6.1: Tabla comparativa de los tres dispositivos. 1 = Mejor, 2 = regular y 3 = peor

Estudiar y analizar los dispositivos nos proporcionó un entendimiento del como funcionan internamente los mismos, lo que ayudó en el diseño de los experimentos para su

evaluación. Comprender el funcionamiento de los dispositivos fue un objetivo vital para la presente tesis, debido a que con este conocimiento pudimos plantear los experimentos, así como utilizar los valores que proporcionan sus sensores en las aplicaciones que se implementaron. Se dedicó un tiempo razonable para lograr una buena comprensión de los tres dispositivos.

En cuanto a los valores del experimento de la aplicación en 2D, podemos ver que el modelo de la Ley de Fitts puede predecir aceptablemente el tiempo que toma desplazar el cursor y seleccionar un objeto. Esto es, el tiempo que toma desplazar el cursor desde un punto de origen determinado hasta realizar la selección de un objeto a una distancia D , es directamente proporcional al índice de desempeño I_d . En otras palabras, mientras más grande es la distancia D y menor el ancho A del objeto, toma más tiempo desplazar el cursor una distancia D hasta el objeto y seleccionarlo.

6.5 Trabajo futuro

Una de las tareas para el trabajo a futuro es la de desarrollar aplicaciones de realidad virtual/aumentada en las cuales se puede interactuar utilizando los tres dispositivos analizados en la presente tesis. Como se pudo observar, en los resultados presentados en la sección 5.3.2 en la página 47, los dispositivos Leap Motion y el Space Navigator presentaron un buen rendimiento en la aplicación en 3D, y el Myo en 2D, así como una fácil navegación e interacción con la misma. Es por esto, que las siguientes tareas son las de explotar el uso de los tres dispositivos en este tipo de aplicaciones, siguiendo las recomendaciones que se mencionan en esta tesis.

Se podría optimizar el uso del Space Navigator, dado que el valor que entrega, w_x , se usa como un peso del desplazamiento máximo, x , entonces el desplazamiento, dx , se calcula como $dx = xw_x$. Entonces se podría obtener el desplazamiento óptimo x para, por ejemplo, optimizar el tiempo de movimiento y su rendimiento.

Algo que no se pudo llevar a cabo en esta tesis y que queda como trabajo a futuro, es la de analizar las propiedades de los sensores del dispositivo Myo. Para esto se tiene que conseguir el equipamiento necesario, tanto para analizar el UMI, como para realizar el análisis de los ocho sensores electromiográficos que contiene el dispositivo.

Un trabajo a futuro interesante es el de proponer un dispositivo basado en control por gestos similar a los tres dispositivos analizados, que tome lo mejor de cada uno. Esto es, hacer un mejor dispositivo basado en los sensores y tecnologías analizadas en la presente tesis.



CÓDIGO PARA USAR EL GIRÓSCOPO DEL MYO COMO RATÓN

El siguiente código se obtuvo de [23] y se modificó para filtrar los datos y agregar el eje z.

```
1 #define _USE_MATH_DEFINES
2
3 #include "MouseMover.hpp"
4
5 #include "Utilities.hpp"
6
7 #include <cmath>
8
9 namespace {
10
11 // For Gain function.
12 // This -may- need to be adjusted based on some system-provided value to handle ←
13 // differences in
14 // resolutions.. but probably not. Leaving this in just in case we need to play ←
15 // with it.
16 // As a fixed value it acts as a constant scaling factor on the gain function.
17 const float pixelDensity = 0.83f;
18
19 // Framerate, though leaving it as-is is likely fine. Ultimately this is just
20 // another number that can be adjusted to achieve desired effect.
21 const float frameRate = 60.0f;
22
23 // These are the constants that characterize the shape of the gain function.
24 // vMax, vMin, CDMax and CDMin have been carefully tuned and should likely be ←
25 // left as they are.
26 // TODO: add reference to the paper that explains the gain function.
27
28 // Input range values
```

```

26 const float vMax = float(M_PI); // 180 degrees in radians
27 const float vMin = 0.174532925f; // 10 degrees in radians
28
29 // Output range values
30 // Sizes in mm of smallest and largest clickable targets,
31 // divided by how much movement is desired for those targets
32 // (for some idealized screen characterized by the pixel density above).
33 const float CDMax = 4580.0f / (float(M_PI) / 6.0f); //45.8 cm / 30 deg in ←
    radians
34 const float CDMin = 16.0f / 0.274532925f; // 16 mm / 15.7 deg in ←
    radians
35
36 // For sensitivity adjustments: Use inflectionRatio and lambda.
37 // For lambda: roughly between 4.0 and 5.0.
38 // A value of 0 will result in constant gain, which may be suitable for gaming.
39 // For inflection ratio: roughly between 0.4 and 0.7.
40
41 // Location of inflection point between vmin and vmax
42 // "Sensitivity", roughly, with a lower inflection ratio yielding more ←
    sensitivity.
43 const float inflectionRatioMin = 0.4f;
44 const float inflectionRatioMax = 0.7f;
45
46 // Slope of curve at inflection point.
47 // Simple proportionality constant.
48 // "Acceleration", roughly.
49 const float lambdaMin = 4.0f / (vMax - vMin);
50 const float lambdaMax = 5.0f / (vMax - vMin);
51
52 // Maps angular velocity to an acceleration curve
53 float getGain(float deviceSpeed, float sensitivity, float acceleration)
54 {
55     // Constant chunk of parameter to exp function below.
56     const float inflectionVelocity = sensitivity * (vMax - vMin) + vMin;
57
58     //gain, in mm/rad
59     float CDGain = CDMin + (CDMax - CDMin) /
60         ( 1 + std::exp(-acceleration * (deviceSpeed - inflectionVelocity)));
61
62     //in px/rad, pixel density scaling factor (px/mm)
63     return CDGain * pixelDensity;
64 }
65
66 float deg2rad(float deg)
67 {
68     return deg * float(M_PI) / 180.0f;
69 }
70
71 } // namespace
72
73 const float MouseMover::defaultAcceleration = 0.3f;
74 const float MouseMover::defaultSensitivity = 0.5f;
75
76 MouseMover::MouseMover()

```

```

77 : _dx(0.0f)
78 , _dy(0.0f)
79 , _dxFractional(0.0f)
80 , _dyFractional(0.0f)
81 , _XTowardsWrist(false)
82 , _acceleration(lambdaMin, lambdaMax, defaultAcceleration)
83 , _sensitivity(inflexionRatioMax, inflexionRatioMin, defaultSensitivity)
84 {
85
86 }
87
88 void MouseMover::onOrientation(const myo::Quaternion<float>& quat)
89 {
90     _quat = quat;
91 }
92
93 void MouseMover::onGyroscope(const myo::Vector3<float> &gyro)
94 {
95     // Accel vector in world space
96     auto gyroRad = myo::Vector3<float>(deg2rad(gyro.x()), deg2rad(gyro.y()), ←
97         deg2rad(gyro.z()) );
98
99     // Gyro vector in world space
100    auto gyroRadWorld = myo::rotate(_quat, gyroRad);
101
102    // Forward vector
103    auto forwardSource = _XTowardsWrist ?
104        myo::Vector3<float>(1.0f, 0.0f, 0.0f) :
105        myo::Vector3<float>(-1.0f, 0.0f, 0.0f);
106    auto forward = myo::rotate(_quat, forwardSource);
107
108    // Right vector
109    auto right = forward.cross(myo::Vector3<float>(0.0f, 0.0f, -1.0f));
110
111    // Get quat that rotates Myo's right vector
112    auto up = myo::Vector3<float>(0.0f, 1.0f, 0.0f);
113    auto yCompensationQuat = myo::rotate<float>(right, up).normalized();
114
115    // Rotate accel vector through y-compensation quat
116    auto gyroVectorCompensated = myo::rotate<float>(yCompensationQuat, ←
117        gyroRadWorld);
118
119    // Get x and y components of accel vector
120    float dx = -gyroRadWorld.z();
121    float dy = gyroVectorCompensated.y();
122
123    // Feed components through the gain function and store.
124    updateMouseDeltas(dx, dy);
125 }
126
127 void MouseMover::updateMouseDeltas(float dx, float dy)
128 {
129     // Get inverse of fps (time)
130     float frameDuration = 1.0f / frameRate; // e.g. 60

```

```
129
130 // Norm of dx + dy
131 float norm = std::sqrt(dx * dx + dy * dy);
132
133 // Get calculated gain based on norm
134 float gain = getGain(norm, _sensitivity.output(), _acceleration.output());
135
136 // Multiply input by gain, returns px (rad/s * px/rad * s)
137 _dx = dx * gain * frameDuration;
138 _dy = dy * gain * frameDuration;
139
140 // Update fractional accumulators
141 _dxFractional += utils::extractFractional<float>(_dx);
142 _dyFractional += utils::extractFractional<float>(_dy);
143
144 // Normalize deltas
145 _dx = std::trunc(_dx);
146 _dy = std::trunc(_dy);
147
148 // Add back in any fractional values that > |1|, adjust fractional ←
    accumulators accordingly
149 if (std::abs(_dxFractional) > 1.0f) {
150     _dx += std::trunc(_dxFractional);
151     _dxFractional = utils::extractFractional<float>(_dxFractional);
152 }
153 if (std::abs(_dyFractional) > 1.0f) {
154     _dy += std::trunc(_dyFractional);
155     _dyFractional = utils::extractFractional<float>(_dyFractional);
156 }
157 }
158
159 void MouseMover::setAcceleration(float value)
160 {
161     _acceleration.setInput(value);
162 }
163
164 float MouseMover::acceleration() const
165 {
166     return _acceleration.input();
167 }
168
169 void MouseMover::setSensitivity(float value)
170 {
171     _sensitivity.setInput(value);
172 }
173
174 float MouseMover::sensitivity() const
175 {
176     return _sensitivity.input();
177 }
```

MouseMover.cpp

CÓDIGO DEL MANEJADOR DEL DISPOSITIVO SPACE NAVIGATOR

El siguiente código muestra la implementación del manejador en espacio del usuario del dispositivo Space Navigator para los sistemas operativos GNU/Linux y Mac OS X, utilizando la biblioteca HIDAPI.

```
1 #ifndef SPACENAV_H
2 #define SPACENAV_H
3
4 #include <stdio.h>
5 #include <wchar.h>
6 #include <string.h>
7 #include <stdlib.h>
8 #include "hidapi/hidapi.h"
9 #include "datamodel.h"
10 #include <qdebug.h>
11 #include <unistd.h>
12
13 namespace spacenav {
14     enum Actions{translation = 1, rotation};
15     enum Buttons{leftButton = 1, righthButton};
16
17     class SpaceNav {
18
19     public:
20         SpaceNav();
21
22         void setUmbralTrans(int umbralTrans);
23         int getUmbralTrans();
24         void setUmbralRot(int umbralRot);
25         int getUmbralRot();
26
```

```

27     static const int POSITIVE = 0;
28     static const int NEGATIVE = 255;
29     static const int constantTrans = 400;
30
31     int result, i;
32     unsigned char buf[256];
33     hid_device *handle;
34     model::DataModel data;
35
36     void sensingData();
37     void analyzeData();
38     void printData(int action);
39
40 private:
41     int umbralTrans;
42     int umbralRot;
43 };
44 }
45
46 #endif // SPACENAV_H

```

spacnav.h

```

1 #include "spacnav.h"
2
3 using spacnav::SpaceNav;
4
5 SpaceNav::SpaceNav() {
6     handle = hid_open(0x046d, 0xc626, NULL);
7     if (!handle) {
8         qDebug("Couldn't open space navigator");
9         return;
10    }
11    // Set the hid_read() function to be non-blocking.
12    hid_set_nonblocking(handle, 1);
13
14    memset(buf,0,sizeof(buf));
15    result = 0;
16    data.setBuf(buf);
17
18    umbralTrans = 30;
19    umbralRot = 5;
20 }
21
22 void SpaceNav::sensingData(){
23     result = hid_read(handle, buf, sizeof(buf));
24     data.reset();
25     data.setRes(result);
26     data.setBuf(buf);
27     data.setAction(buf[0]);
28     analyzeData();
29 }
30

```

```

31 void SpaceNav::analyzeData(){
32     int tmp;
33     if(data.getRes() == 3){//Buttons pressed
34         data.setButton(buf[1]);
35         if(buf[1] == leftButton){
36             data.setLeftBtn(1);
37         }else if(buf[1] == righthButton){
38             data.setRightBtn(1);
39         }
40     }else if(data.getRes() == 7){
41         if(data.getAction() == translation){//Translation 3 degrees of freedom
42             //In X axis (left - right)
43             switch(buf[2]){
44                 case POSITIVE:
45                     tmp = buf[1];
46                     if(tmp > umbralTrans){
47                         data.setX(tmp - umbralTrans);
48                     }
49                     break;
50                 case NEGATIVE:
51                     tmp = buf[1] - NEGATIVE;
52                     if(tmp < -umbralTrans){
53                         data.setX(tmp - umbralTrans);
54                     }
55                     break;
56                 case NEGATIVE - 1:
57                     data.setX(-constantTrans - umbralTrans);
58                     break;
59                 case POSITIVE + 1:
60                     data.setX(constantTrans - umbralTrans);
61                     break;
62             }
63
64             //In Y axis (down - up)
65             switch(buf[6]){
66                 case POSITIVE:
67                     tmp = buf[5];
68                     if(tmp > umbralTrans){
69                         data.setY(tmp - umbralTrans);
70                     }
71                     break;
72                 case NEGATIVE:
73                     tmp = buf[5] - NEGATIVE;
74                     if(tmp < -umbralTrans){
75                         data.setY(tmp - umbralTrans);
76                     }
77                     break;
78                 case NEGATIVE - 1:
79                     data.setY(-constantTrans - umbralTrans);
80                     break;
81                 case POSITIVE + 1:
82                     data.setY(constantTrans - umbralTrans);
83                     break;
84             }

```

```

85
86 //In Z axis (back - forward)
87 switch(buf[4]){
88     case POSITIVE:
89         tmp = buf[3];
90         if(tmp > umbralTrans){
91             data.setZ(tmp - umbralTrans);
92         }
93         break;
94     case NEGATIVE:
95         tmp = buf[3] - NEGATIVE;
96         if(tmp < -umbralTrans){
97             data.setZ(tmp - umbralTrans);
98         }
99         break;
100    case NEGATIVE - 1:
101        data.setZ(-constantTrans - umbralTrans);
102        break;
103    case POSITIVE + 1:
104        data.setZ(constantTrans - umbralTrans);
105        break;
106    }
107    //qDebug(" %d %d %d %d %d %d %d %d", buf[0], buf[1], buf[2], buf[3], ←
108        buf[4], buf[5], buf[6], buf[7]);
109 }else if(data.getAction() == rotation){//Rotation 3 degrees of freedom
110 //In X axis (pitch)
111 switch(buf[2]){
112     case POSITIVE:
113         tmp = buf[1];
114         if(tmp > umbralRot){
115             data.setX(tmp - umbralRot);
116         }
117         break;
118     case NEGATIVE:
119         tmp = buf[1] - NEGATIVE;
120         if(tmp < -umbralRot){
121             data.setX(tmp - umbralRot);
122         }
123         break;
124     case NEGATIVE - 1:
125         data.setX(-constantTrans - umbralRot);
126         break;
127     case POSITIVE + 1:
128         data.setX(constantTrans - umbralRot);
129         break;
130     }
131 //In Y axis (yaw)
132 switch(buf[6]){
133     case POSITIVE:
134         tmp = buf[5];
135         if(tmp > umbralRot){
136             data.setY(tmp - umbralRot);
137         }

```

```

138         break;
139     case NEGATIVE:
140         tmp = buf[5] - NEGATIVE;
141         if(tmp < -umbralRot){
142             data.setY(tmp - umbralRot);
143         }
144         break;
145     case NEGATIVE - 1:
146         data.setY(-constantTrans - umbralRot);
147         break;
148     case POSITIVE + 1:
149         data.setY(constantTrans - umbralRot);
150         break;
151     }
152
153     //In Z axis (roll)
154     switch(buf[4]){
155     case POSITIVE:
156         tmp = buf[3];
157         if(tmp > umbralRot){
158             data.setZ(tmp - umbralRot);
159         }
160         break;
161     case NEGATIVE:
162         tmp = buf[3] - NEGATIVE;
163         if(tmp < -umbralRot){
164             data.setZ(tmp - umbralRot);
165         }
166         break;
167     case NEGATIVE - 1:
168         data.setZ(-constantTrans - umbralRot);
169         break;
170     case POSITIVE + 1:
171         data.setZ(constantTrans - umbralRot);
172         break;
173     }
174 }
175 }
176 }
177
178 void SpaceNav::printData(int action) {
179     if(action == rotation){
180         qDebug("rx = %2f, ry = %2f, rz = %2f", data.getRx(), data.getRy(), ←
181             data.getRz());
182     }else if(action == translation){
183         qDebug("x = %2f, y = %2f, z = %2f", data.getX(), data.getY(), ←
184             data.getZ());
185     }
186 }
187
188 void SpaceNav::setUmbralTrans(int umbralTrans){
189     this->umbralTrans = umbralTrans;
190 }

```

```
190 void SpaceNav::setUmbralRot(int umbralRot){
191     this->umbralRot = umbralRot;
192 }
193
194 int SpaceNav::getUmbralTrans(){
195     return umbralTrans;
196 }
197
198 int SpaceNav::getUmbralRot(){
199     return umbralRot;
200 }
```

spacenv.cpp

BIBLIOGRAFÍA

- [1] Frank Weichert, Daniel Bachmann, Bartholomaus Rudak and Denis Fisseler. Analysis of the accuracy and robustness of the leap motion controller. *Sensors*, 13:6380–6393, 2013.
- [2] Jože Guna, Grega Jakus, Matevž Pogačnik, Saso Tomažič and Jaka Sodnik. An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking. *Sensors*, 14:3702–3720, 2014.
- [3] Accesorios para wii. <https://www.nintendo.es/Wii/Accesorios/Accesorios-Wii-Nintendo-Ib-eacute-rica-626430.html> (consultado el 18 de Septiembre del 2015).
- [4] Kinect. <http://www.xbox.com/es-MX/Kinect> (consultado el 18 de Septiembre del 2015).
- [5] Sushmita Mitra and Tinku Acharya. Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics-part C: Applications and Reviews*, 37:311–324, 2007.
- [6] Soren Lenman, Lars Bretzner and Bjorn Eiderback. Computer vision based recognition of hand gestures for human-computer interaction. ISSN 1403 - 0721, 2002.
- [7] S Mann. Intelligent image processing. John Wiley & Sons, 2001.
- [8] Moniruzzaman Bhuiyan and Rich Picking. Gesture-controlled user interfaces, what have we done and what's next? *In Proceedings of the Fifth Collaborative Research Symposium on Security, E-Learning, Internet and Networking (SEIN 2009)*, pages 59–68, 2009.
- [9] Gestureteek. <http://www.gesturetekhealth.com/products-rehab.php> (consultado el 18 de Septiembre del 2015).
- [10] Rastislav Lipovský and Hugo Alexandre Ferreira. Hand therapist: A rehabilitation approach based on wearable technology and video gaming. *Bioengineering (ENBENG), 2015 IEEE 4th Portuguese Meeting on*, pages 1–2, 2015.
- [11] Ahmad M Qamar, Ahmed Riaz Khan, Syed Osama Husain and Md. Abdur Rahman. A multi-sensory gesture-based occupational therapy environment for controlling home appliances. *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 671–674, 2015.

- [12] Kristian Nymoen, Mari Romarheim Haugen and Alexander Refsum Jensenius. Myo - evaluating and exploring the myo armband for musical interaction. *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 215–218, 2015.
- [13] Ivan Phelan, Madelynne Arden, Carol Garcia and Chris Roast. Exploring virtual reality and prosthetic training. *IEEE Virtual Reality Conference 2015*, pages 353–354, 2015.
- [14] Giulio Marin, Fabio Dominio and Pietro Zanuttigh. Hand gesture recognition with leap motion and kinect devices. *Image Processing (ICIP), 2014 IEEE International Conference on Paris*, pages 1565–1569, 2014.
- [15] Frederic Kerber, Pascal Lessel and Antonio Kruger. Same-side hand interactions with arm-placed devices using emg. *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, pages 1367–1372, 2015.
- [16] Pongphan Pongpanitanont and Warakom Charoensuk. Leap motion@: Signal preservation and medical training system. *Biomedical Engineering International Conference (BMEiCON)*, pages 1–4, 2014.
- [17] Daniel Bachmann, Frank Weichert and Gerhard Rinkebauer. Evaluation of the leap motion controller as a new contact-free pointing device. *Sensors*, 15:214–233, 2015.
- [18] Alena Petrasova, Julie V. Farrer, Silvester Czanner, Alan Chalmers and Dieter Wolke. User interface for assisting babies with feeding disorders. *International Conference on CyberWorlds*, pages 287–291, 2009.
- [19] Daniel López Escogido. *Escáner 3D de Alta Precisión*. PhD thesis, Centro de Investigación y de Estudios Avanzados del IPN, Ciudad de México, 6 2016.
- [20] What is metrology. <http://www.bipm.org/en/worldwide-metrology/> (consultado el 07 de Marzo del 2016).
- [21] Indian standard. manipulating industrial robots. Performance criteria and related test methods; (first revision), 2005.
- [22] Jack Jeswiet and Ray Helferty. Measuring robot repeatability an application of ISO and ANSI standards. *Advanced Robotics*, 10:503–520, 1995.
- [23] Build your own mouse control with myo. <http://developerblog.myo.com/build-your-own-mouse-control-with-myo/> (consultado el 17 de Agosto de 2016).
- [24] Steven W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, San Diego, California, 1997.
- [25] Myo coordinate system. <https://developer.thalmic.com/forums/topic/41/> (consultado el 17 de Agosto de 2016).
- [26] 3dconnexion. <http://www.3dconnexion.com> (consultado el 02 de Agosto de 2016).

-
- [27] Jan Axelson. *USB Complete: The Developer's Guide*. Lakeview Research, 4 edition, 2009.
- [28] HID API for Linux, Mac OS X, and Windows. <http://www.signal11.us/oss/hidapi/> (consultado el 14 de Septiembre de 2016).
- [29] R. William Soukoreff, I. Scott MacKenzie. Towards a standard for pointing device evaluation, perspectives on 27 years of fitts' law research in hci. *International Journal of Human-Computer Studies*, 61:751–789, 2004.
- [30] Iso 9241-9. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=30030 (consultado el 17 de Agosto de 2016).
- [31] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47:381–391, 1954.
- [32] Alexandros Pino, Evangelos Tzemis, Nikolaos Ioannou, Georgios Kouroupetroglou. Using kinect for 2d and 3d pointing tasks: Performance evaluation. *Human-Computer Interaction. Interaction Modalities and Techniques*, 8007:358–367, 2013.
- [33] I. S. MacKenzie. A note on the validity of the Shannon formulation for Fitts' index of difficulty. *Open Journal of Applied Sciences*, 3:360–368, 2013.
- [34] Heiko Drewes. A lecture on Fitts' Law, 2013. [http://www.cip.ifi.lmu.de/drewes/science/fitts/A lecture on Fitts Law.pdf](http://www.cip.ifi.lmu.de/drewes/science/fitts/A%20lecture%20on%20Fitts%20Law.pdf).
- [35] Alberto Pretto and Giorgio Grisetti. Calibration and performance evaluation of low-cost imus. *20th IMEKO TC4 International Symposium and 18th International Workshop on ADC Modelling and Testing*, 2014.